DARWARS Architecture

A BBN Technologies Document

Version 2.0

31 January 2005

Please e-mail comments to: darwars@bbn.com

A copy of this document will be available online at: https://www.darwars.net

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
31 JAN 2005	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
DARWARS Architecture: A BBN Technologies Document	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Office of Naval Research 800 North Quincy Street Arlington, VA 22217-5660**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**72**	

Document History

29 October 2003	Initial Publication (v1.0)
31 January 2005	Publication of v2.0

Change Log

Date	***Section***	***Change***
31 January 2005	All sections	Minor edits
		Last Meter Training Systems ➔ Training Systems
		Updated figures 3,4,5,8
	(former) 4.2.2	Section on “Database Middleware” deleted.
	(former) 4.2.7	Section on “Other Data Services” deleted.

Contents

Figures

Tables

This page is blank intentionally.

1 Introduction

1.1 The DARWARS Program and Vision

U.S. military personnel who emerge from the Combat Training Centers (CTC) are the best trained in the world. DARWARS aims to bring the level of excellence achieved at the CTCs to all our forces, all the time, everywhere and to do so at a lower cost. BBN will achieve this vision with the creation of DARWorld – an innovative training environment that encompasses communities of members including trainees, instructors, subject matter experts, and authors of training content.

In this document, DARWARS is the DARPA program integrating training components, tools, and infrastructure into DARWorld, which is a distributed training system[1] consisting of many interconnected components and applications ranging from single-user training systems to multi-user and massively multi-user training systems to authoring and administrative tools.

The five key characterizations or requirements of DARWorld are:

- Universal – available to individuals, teams of individuals, and teams of teams and including instructors and others concerned with the training of our armed forces.
- Continuous and Persistent –available 24 by 7 with training relevant to the individual and team based on training objectives and the progress already made toward those objectives.
- On Demand – available without elaborate preparation.
- Engaging – trainees will seek out training opportunities instead of simply meeting requirements.
- Effective – training packages and training results linked to training objectives paces training to achievement.

1.2 The DARWARS Approach

Addressing the five requirements above leads to a system with two focal points. First is a set of effective and engaging training systems that embody specific knowledge and instructional expertise, and that establish and guide the training scenario to create events to challenge the students and monitor the students' performances. Intelligent tutoring and other techniques enhance the training experience by offering advice or comment during or after the training session.

The second focus is an infrastructure to bring the benefits of these training systems to all DARWorld members wherever they might be and whenever they might choose to use them. The infrastructure, with its backend databases and associated servers, provides the engaging and effective distribution of the training system to a wide universe of individuals and teams continuously, persistently, and on demand.

Pedagogical ontologies (Figure 1) serve as the bridge between individuals or teams and training packages designed to promote specific competencies. Training objectives, training packages, and training results are linked together by these ontologies.

---

[1] Previous versions of this document referred to "last meter training systems" (LMTS) meaning training systems running on ordinary desktop and laptop computers. This terminology is obscure and requires explanation. Since there is no inherent reason limiting the DARWARS architecture to such systems, the "last meter" qualification has been removed throughout this document and the simpler, more encompassing "training system" used instead.

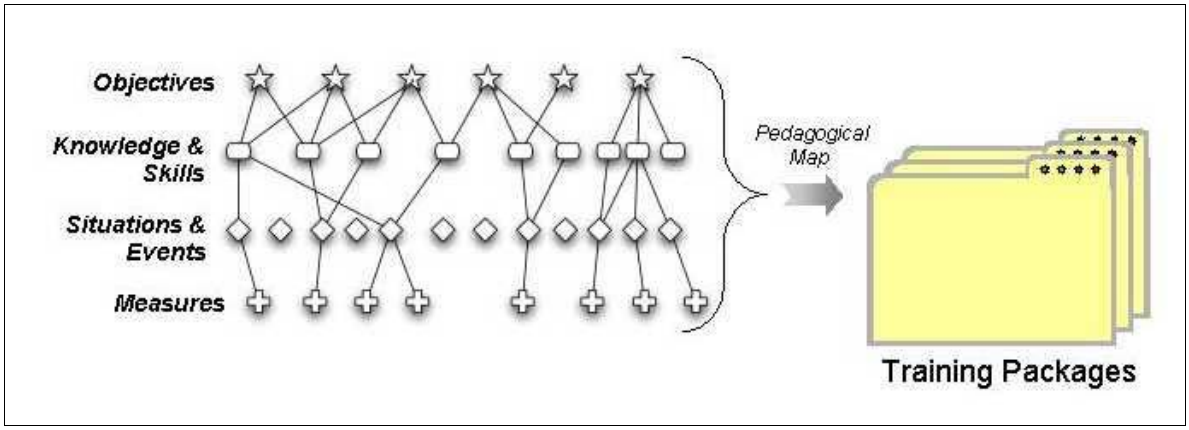

Figure 1: DARWorld Pedagogical Ontologies

1.3 DARWorld Architectural Issues and Choices

DARWorld will be ubiquitous and persistent; it will provide training wherever and whenever desired. This leads to an overall architecture that employs a backend of servers that maintains and distributes information through the Internet to wherever a user might be located whenever he chooses to use it. This backend can be both centralized and distributed. It can be centralized for ease of maintenance and tight integration of certain components and distributed to improve robustness in the face of localized failures.

To be effective, DARWorld must be easy and friendly to use. This leads to the choice of applications with which the user is already familiar wherever possible. The user can use a web browser for much of the administrative activity and instant messaging and email are used for routine communication and for scheduling and arranging for training sessions.

DARWorld will install and maintain software on the client machines with a high degree of automation up to a limit the user can select. Only a minimal set of software must be installed in the conventional way for DARWorld to bootstrap its way onto a user's machine. This means a user using new DARWorld features only needs to, at most, approve the installation of the new or upgraded software. He can even choose to have software installed without approval with the only noticeable difference being some progress messages as the upgrades are performed.

The database requirements of the servers will be aligned to avoid the disconnect that frequently happens when traversing from one system to another; a DARWorld user will *not* be asked to login more than once and, in many cases, he will not be asked at all; his security token will serve to identify him[2]. (He may have to supply a password to activate the token.)

DARWorld will use several standards, not so much for interoperability reasons, though that is a concern, but more for the savings possible by not re-inventing the wheel. E.g., Section 11.1 lists several learning management standards. These and other standards are mentioned in context elsewhere.

[2] The use of a hardware security token is not without its drawbacks. Chief among these is that the user must remember to take his security token with him to prevent another user from walking up to his machine and impersonating the first user. The software alternative has the risk of exposure in moving keys to another machine or the inconvenience of obtaining another certificate for a different machine.

While DARWorld is mainly about training, there are several other roles that must be filled to create a useful system. DARWorld addresses this issue by specifically separating the training needs of its users (members) from other information. This same principle is applied for other classes of data that will typically apply only to a subclass of the users. Similarly, some of the information that applies to certain users also applies to teams or groups of users. So the concept of DARWorld members is expanded to include groups and other entities not normally considered to be "users".

DARWorld services will be specified in terms of the protocols used to access them and reference implementations in Java will be created that can be used on multiple platforms for those applications that are also written in Java. This does *not* mean applications must be written in Java. DARWorld will coordinate the development of implementations in other languages to minimize the duplication of effort and help insure their correctness.

1.4 Guide to this Document

1.4.1 Goals and Scope of this Document

This living document describes an evolving vision of the architecture required to support the goals of the DARWARS program. We offer a framework for the future evolution of DARWARS, promoting a graduated integration that allows training systems to leverage a distinct set of capabilities and to participate in the ubiquitous, persistent DARWorld.

This document will be refined throughout the DARWARS program, reflecting the results of training system development, component development, technology integration experiments, and evolving COTS/GOTS technologies. The goals of this document are to:

- Present a comprehensive long-term vision of the DARWorld architecture to the government sponsors, potential DARWARS partners, DARWARS training system developers, and component developers: its requirements, services and capabilities, benefits, design, and open issues.

- Provide a blueprint for the short- and mid-term development of the DARWorld infrastructure and the integration of training systems and components into DARWorld.

- Serve as a common basis and reference for further DARWorld specifications, such as a Developers' Guide, Users' Guide, or Integration Plan.

- Clarify the degrees of freedom in the design of DARWorld, and point out design choices, tradeoffs, and alternatives. There are design features that we recommend, some that we slightly favor, and some that are to be determined.

According to good software engineering practice, this document describes the design from various perspectives, providing different angles on the architecture. The main views are the component view describing the elements of the distributed infrastructure, the interface view discussing the hooks into the infrastructure, and the functional view presenting the capabilities of the systems implemented by the interaction of its components. In addition, the use case view provides a cross-reference to all other views.

There are two levels of architecture within DARWorld that are both covered by this document:

- The distributed DARWorld application. This is the overall distributed system including all the training systems, tools, and components, and how they interact to accomplish the overall functionality.

- The individual application being part of / participating in DARWorld. This is a single training system or training tool that uses a number of DARWorld features in order to participate in DARWorld.

While discussing the scope of this architecture, it is important to say what is not covered by this document: there are no detailed APIs given, no specifications for tool design, and no design decisions that are either not architectural or very low level. Future specification efforts under the DARWARS program will address these aspects.

Finally, a prominent question regarding the architecture design is the time horizon of this architecture, or of the underlying DARWARS vision. As mentioned earlier, this vision is evolving and will be influenced by both the direction given by DARPA and the reactions of the various Services interested in DARWorld. The concrete steps towards an implementation are laid out in the DARWARS integration plan that is referenced in Section 12. However, we believe that the fundamentals of the architecture described in this document will serve as a solid basis for a generation of DARWorld implementations.

1.4.2 Intended Audience

This document is initially intended for the DARWARS program management team, the advisors to the government, and all DARWARS contractors such as training system and component developers. It is assumed that future versions of this document will be released to a large and diverse audience. Expected readers include military and commercial training system developers, developers of training system components, educational and content developers, and others.

1.4.3 Document Organization

This document presents different views/perspectives/angles on the DARWorld architecture and is organized as follows:

Section 1 is this Introduction.

Section 2, Architecture Overview, provides an introduction to our solution to the DARWARS problem and presents the principal components and relationships in DARWorld, the name we are giving to the virtual learning environment described in this document.

Section 3, DARWorld Functional View, describes the architecture from a functional perspective and lays out its functional elements as well as the services (and their taxonomy) that support it.

Section 4, Distributed DARWorld Infrastructure, addresses the architecture from a distributed component view. It explains what the different servers/daemons/clients/peers are and how they are interconnected.

Section 5, Generic Training System Architecture, provides a closer look into the architecture of a generic training system and its interfaces, and discusses how training systems will interoperate in DARWorld.

Section 6, Infrastructure Interfaces, presents the “hooks” into the infrastructure that are used by training systems and other tools that want to be integrated and participate in DARWorld.

Section 7, Use Cases, looks at DARWorld from the use case perspective. A few examples of the most significant use cases are discussed including their motivating vignettes as well as the corresponding sequence of actions performed by the infrastructure. Note that some readers, especially those not familiar with the domain of military training, may find it beneficial to read this section first before looking at the architectural details.

Section 8, Data Management Design, contains the view on data relevant to DARWorld and its operation. This includes data types, persistence issues, and security aspects of the system.

Section 9, DARWorld Administration, discusses issues regarding system administration, such as user management, bug reporting, and error handling.

Section 10, Conclusions, summarizes this document and discusses various design choices and tradeoffs.

Section 11, Appendix, contains a section on e-learning initiatives, and a Glossary that explains the fundamental DARWARS/DARWorld terminology.

Section 12, References, lists additional references and useful links to background information and other documents related to the development under DARWARS.

1.4.4 Acknowledgements

This document has been produced by the BBN architecture team. Many valuable inputs to this document have come from discussions with the training system and component developers (CHI Systems, ISI, Acuitus). We would like to thank them for their contributions of ideas on architecture, infrastructure, and cross-cutting components. Special thanks go to DARPA PM, Dr. Ralph Chatham, for his overarching vision and guidance, and to the rest of the government team: Dr. Harold Hawkins, Ray Perez, Dexter Fletcher, J.C. Herz, Jason Robar, Avron Barr. We also benefited from informative and insightful conversations with many other individuals: Philip Dodds, Michael Zyda, Harry O'Neil, and many others.

2 Architecture Overview

2.1 Primary Design Principles and System Characteristics

The DARWorld architecture is founded on a small number of principles. These principles are intended to lead to a more easily understood design that retains desirable properties of performance and reliability. One of these principles is *parsimony of concepts*, meaning we keep the number of concepts that need to be handled and understood to a minimum. We describe operations in terms of the most general concepts possible and reserve more specific concepts for the few cases that require them.

A second principle is *consistency*, meaning that, wherever possible, similar problems are solved in similar ways. For example, we do much of the distributed communication with the same protocol; only when there is a strong reason do we use a different protocol for a particular kind of communication. Naturally, the details vary because the data varies, but at a high level, the protocols are the same.

A third principle is to use *open standards* and *open source software* wherever possible. There is no reason to recreate software and re-solve the same problems that have already been addressed by others. At the same time, we cannot afford to be locked out of adapting those solutions to the particular requirements of DARWorld because they are proprietary. Open standards enhance the prospect of finding open source software and open source software is more amenable to adaptation than proprietary solutions.

The specification for DARWorld has the following system characteristics:

- *Distributed Processing* – DARWorld is a distributed system to provide a scalable and cohesive training experience to multiple users at many locations at arbitrary times.

- *Flexibility of Communications* – efficient use of communications bandwidth leads to highly encoded and compact data representations. Such representations are often brittle and difficult to debug. Less efficient representations are easier to debug and bend to evolving requirements. DARWorld communications leans toward the latter, but employs more efficient representations when performance is likely to be an issue.

- *Reusability* – training systems should be constructed in a way to promote reuse of their components. The infrastructure design will promote reuse in training systems and other tools.

- *Deployability* – the architecture allows for operation on portable equipment, potentially at remote sites with poor or intermittent connections to the main DARWorld servers and services.

- *Usability* – the DARWorld architecture has a number of features that facilitate the creation of easy-to-use user interfaces. Examples include the DARWorld Object Reference, a common database for all applications (DARWorld knows who I am), and automatic software install/upgrade capabilities.

In the following section, we will describe the significant parts of the DARWorld architecture and how we composed a solution adhering to the principles above.

2.2 System Description

DARWorld is designed to provide, to teams of users, rich shared training experiences on-demand, the ability to rehash and revisit these experiences, and the means to modify and create new ones with ease. Some experiences may persist for long periods of time, others for only a session; some may involve a single user working alone (possibly in concert with simulated fellow participants), others may involve units and larger groupings of participants in combined operations; some may include instructors in the loop to create, guide and evaluate the experience, others may occur without any human scrutiny or intervention; some may occur spontaneously with teams assembled from available players, others may be carefully scheduled in advance for established teams of players. DARWorld aims to make available a rich palette of experiences and the social and instructional frameworks that promote effective training.

As a system of systems, DARWorld provides

- the framework for integrating current and future simulation based training environments and their associated instructional components
- a "software house" for components and tools to simplify building new environments, scenarios and instruction
- a software management substrate to deliver and maintain client software
- data collection and storage facilities for the review, annotation, and analysis of events
- and the communication tools to weave a strong social fabric around the training that DARWorld provides.

What is being "integrated" in DARWorld?

We choose to think of the training system as the building block of DARWorld. We view each training system as a source of experience for a user, as the source for instruction about particular skills and knowledge, and as the primary interface to the simulated world. This basic high-level breakdown of training system functionality into Reality Engine, Instructional Component, and User Interface, respectively, is depicted in Figure 2, which shows the distribution of responsibilities among the components and the types of communication between them. This is a simplified view of the training system from the user's perspective, whose experience in a session is governed by the functionality of the client software that manages these three aspects of interaction with DARWorld. In fact, these components may be coupled to the training system clients of other users. For example, if the reality engine of a training system is something like the Unreal engine or a MMP game engine, there will be a server cluster supporting a shared reality among the participants. Similarly, in a team trainer, there will be a shared instructional component that provides coaching and assessment and AAR for groups of players.

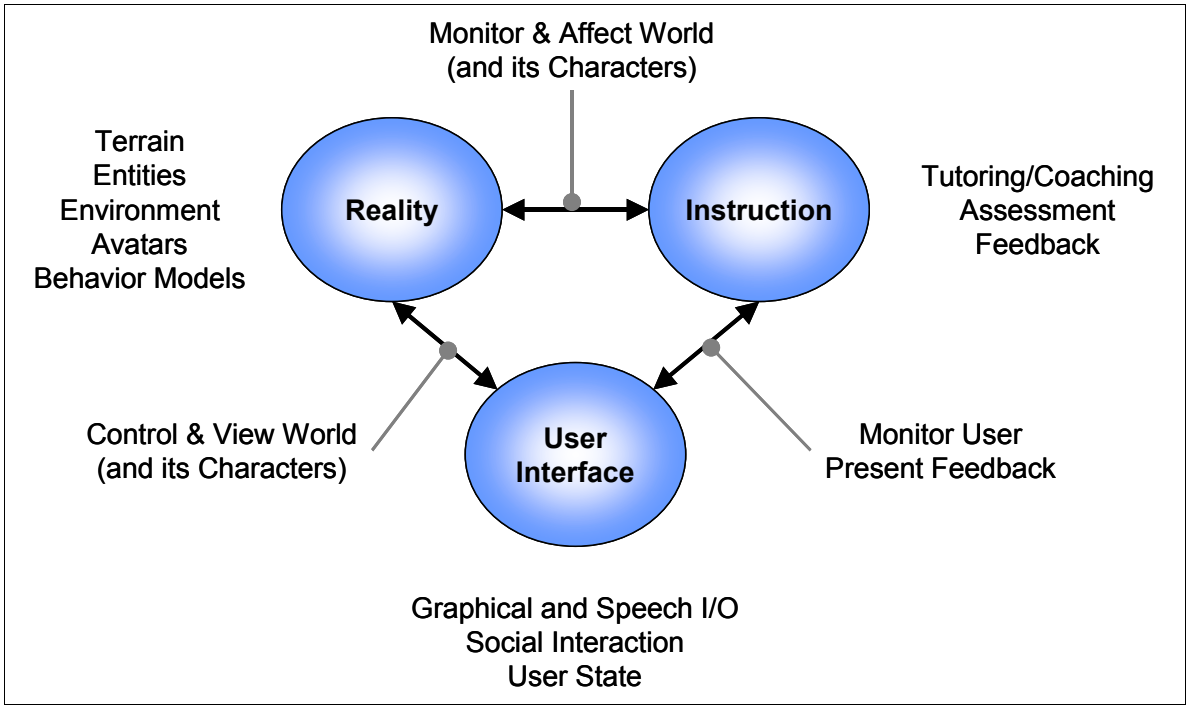

Figure 2: High-level Breakdown of Training System Functionality

It is possible that a training system has no instructional component, as would be the case of a typical simulation-based trainer that provided an interface to a shared reality but relied on human instructors for feedback. It is possible that a training system would have no reality engine, if it were based entirely on pre-prepared materials. In the case of Acuitus' IT trainer, the reality is provided by actual hardware. It is at least conceivable that a training system has no user interface, as would be the case of an instructional component controlling the behavior of NPCs in a shared world. These special cases aside, the view of the training system as the window onto the learning experience for each user has proven a useful one for organizing our thinking about the role of the architecture in DARWorld.

The architecture places no restriction on the scale of the experience – how many individuals are involved. It is meant to encompass the set of training system currently under development in the DARWARS program, and to accommodate training systems that are focused on a single user with the entire interaction under the control of the training system (e.g., the Language Tutor) as well as training systems that involve many participants in less predictable situations (e.g., the Air Mission Trainer, HLA Federations, MMOGs).

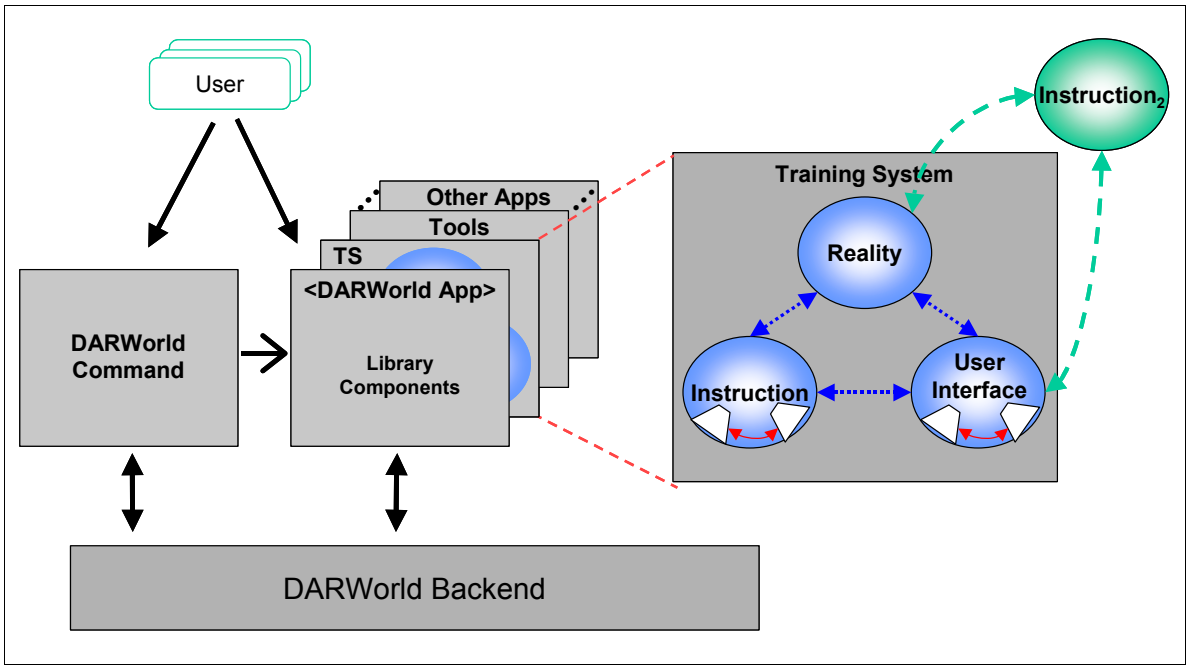

Figure 3: DARWorld Architecture Overview: Training System Role Within the Infrastructure

How are training systems integrated in DARWorld?

Figure 3 shows the major elements of the DARWorld framework that together provide infrastructure and services for training systems as well as support additional interactions with DARWorld that are not exclusively tied to having the experience provided by a training system (e.g., lobby functions).

Libraries & Components – A set of software components that are either linked as libraries or used to invoke services from within a training system. Examples are translation libraries, IM, or VoIP clients that could be embedded in the user interface component of a training system. Some of these components require server support, which would be part of the DARWorld shared infrastructure.

DARWorld Applications – Besides the various training systems, there is a collection of software to handle interactions with users (trainees, trainers, administrators, authors, observers, controllers) outside of the context of a particular training system. In general these provide social, management, and authoring functions of DARWorld. Support for social interaction takes many forms: scheduling events, role matching and team formation, sending instant messages, and web publishing, are some. Management functions include, for example, personal identity and profile management, scenario searching and launching, client code configuration management, preparing for disconnected operation, detecting and dealing with unresponsive nodes in the network, searching for scenarios that meet criteria for addressing particular competencies or requiring certain roles and skill levels. Authoring support includes scenario descriptions, individual and team competencies, and web materials. Many of these applications will in fact be Web-based – accessed via a browser.

A special case of a DARWorld application is the "DARWorld Command" that will be explained in detail in Section 3.1.

DARWorld Backend – The backbone of DARWorld is one or more servers to support sharing, distributed operation, and persistent distributed data storage. Data include record keeping (individual and group identities and profiles), scenario content, data collected for analysis, AARs and portfolios kept by users. All components of a training system will rely on these servers. Some servers support the simulation environment used by the training system. Servers are available to store user modeling data, support communications with other users during play, store users profiles, assessments, and AARs. The backend servers support universal references to objects in DARWorld – a mechanism by which scenarios (and events within scenarios) can be shared between users (and applications) as a URL that can be included in web pages, email, or messages.

The DARWorld architecture allows temporary disconnected operation in which groups of users, while connected to each other, are not part of the larger DARWorld network. Support will be provided for packaging software and data needed for remote operation and for rejoining the DARWORLD network and merging data collected while operating remotely.

Combined training system operation – Figure 3 shows a second instructional component, I_2, combined with the primary training system to lend its expertise to the instruction available for a user. The component view of training system architecture led us to consider a form of training system interoperability distinct from simulation interoperability (i.e., between reality engines). The goal of this piece of the architecture is to allow instructional components originally built as part of all-inclusive training system to be reapplied in other situation, e.g., a MMPG, where they would be able contribute to the instructional palette available in this new setting. (For more details, please see Section 5.)

2.3 DARWorld Architecture Support for Desired Capabilities

The following table shows some of the desired capabilities pertaining to DARWorld and how the DARWorld architecture will support them. The capabilities of DARWorld will evolve as developers contribute components, which over time may become higher-level DARWorld services. In the remainder of the document, the various DARWorld functionalities will be explained in more detail.

Table 1: Desired Capabilities and Supporting DARWorld Functionalities

Capability	***Supporting DARWorld Functionality***
Continuously Available Training • **Deployable Training:** on-demand access to individual and team training experiences • **Common repository for user data:** Stored profiles and portfolios for individuals and teams	• **Collaboration Tools** • **Matching Services** (Roster; Training Package) • **Disconnected Operation (and merging)**
Support a variety of classes of users and use cases: • **Supports many learning formats**: able to support the standard e-learning formats, computer-based instruction mechanisms, and intelligent tutoring capabilities. • **Supports a variety of ways to author content**	• **SCORM compliant content** can be subsumed and made available. • **Generic training system architecture** that does not restrict the types of training systems that can be constructed within DARWorld.
Training System Development & Reconfiguration • **Component Libraries**: components capabilities, needs, and interfaces are available to developers. • **Component Interoperability**: independently developed and deployed systems seamlessly work together and communicate at varying levels of semantics. • **Component Reuse**: use component at sites or in test or training events other than those in which it was originally designed to operate. • **Component Adaptability**: components can adapt to other components.	• **Metadata Directory** services to register components and their capabilities & interfaces • **Ontology services** to describe capabilities • **Communication mechanisms and infrastructure** to exchange messages across the network • **Variable levels of semantics** for communication and training content using common protocols, standards, and ontologies • **Translation services** to support interoperability
Transformation and Evaluation • **Reliability and Security**: applications are fault tolerant and secure. They can recover from failures.	• **Exception management services, component lifecycle management services**, **and security services** for reliable and secure operation • **Logging and event services** for capturing and recording user activity • **Visualization services** for human visualization and control of training activity.
Distributed Team Training • **Scalability:** simultaneously training teams and many individuals on many levels	• **Team coordination services** to help ensure efficient component interaction on tasks and support dynamic, adaptable teams • **Collaboration Tools**
Easy and Fast to Build, Low Cost to Develop • **Componentized:** Applications can be composed using a number of preexisting components and a generic architecture and infrastructure • **Programmability:** programmers can easily make components ready to participate in applications.	• **Infrastructure services and adapters** for connecting components • **Logging, event, visualization, simulation, and debugging services** to instrument, visualize, and debug a application • **Policy and protocol management services** to customize applications

Capability	*Supporting DARWorld Functionality*
• **Adaptability/Customizability**: applications can be customized for a particular use or domain through content changes with little or no changes to the application itself. • **Testability**: components and applications can be easily tested and debugged.	• **DARWorld Management Services** to start, monitor, manage, and maintain DARWorld services and infrastructure

3 DARWorld Functional View

The DARWorld architecture is a collection of infrastructure, services, components, standards, and protocols that enables the integration, interoperability, and deployability of training applications and software components as part of DARWorld, as described in Section 2.

This section describes the desired capabilities of the DARWorld architecture, its functional elements, and the services and components that support it.

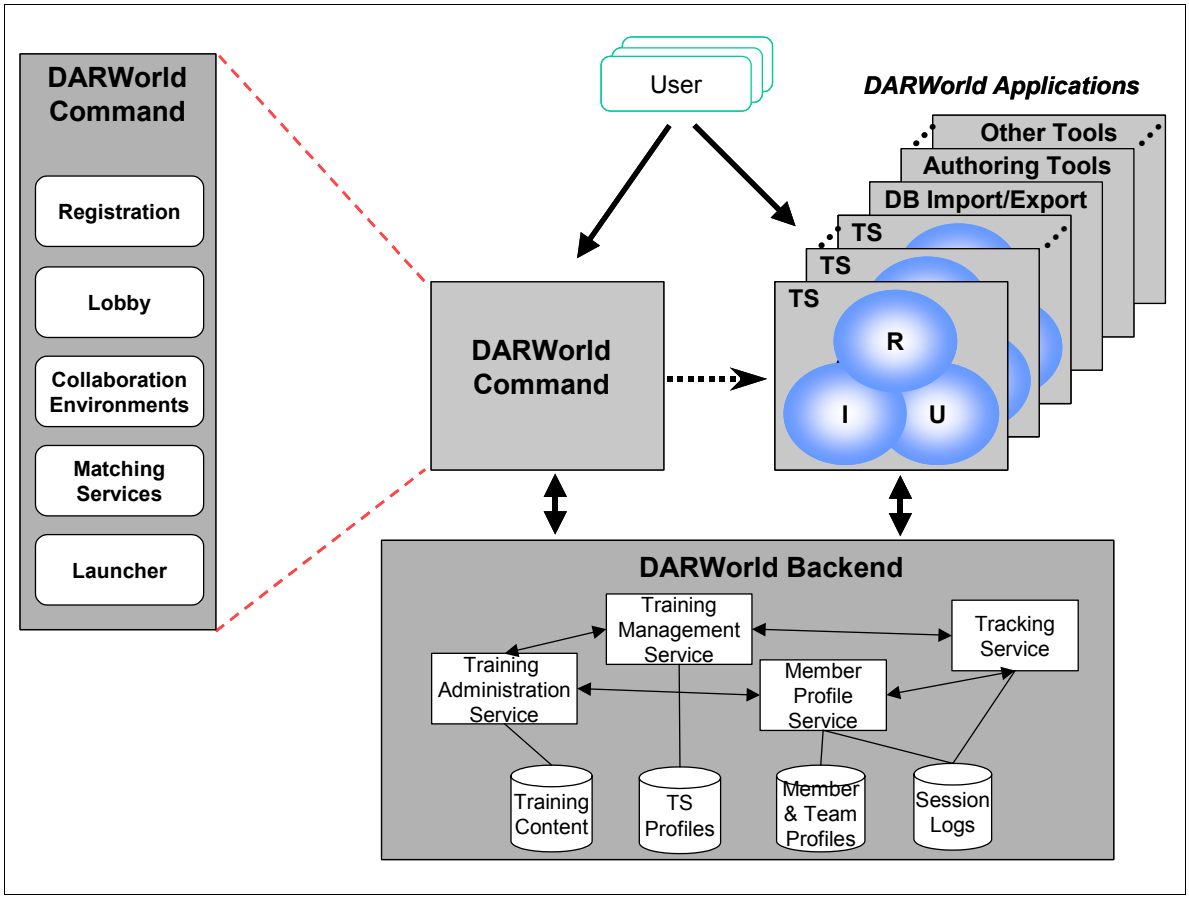

Figure 4: DARWorld Architecture Overview: Functional View

3.1 DARWorld Command

The DARWorld Command application provides the main entrance for a user to the DARWorld system. This platform provides the user interfaces required to organize a training event, launch a training event and follow it up with After Action Review (AAR), feedback and suggestions. These actions invoke separate applications (such as the training system) to complete the action. To complete those tasks, the capabilities outlined in the following sections are necessary parts of the DARWorld Command application.

3.1.1 Registration

Registration

Registration provides entrance into the DARWorld system, allowing access to the DARWorld Lobby, through which a user can access training offerings and content, collaboration tools, and training applications. Users could be registered automatically while navigating a DARWorld portal or explicitly directed to register upon accessing DARWorld. Registration manages the user's access to training offerings and history, learning plans, and user schedules.

Personalization

A user's experience (and training needs) is produced using a combination of the user's system profile (status, access privileges, etc.) and personal preferences. Member preferences may be edited by the user and can be used to control features such as user presence status (e.g., online, free to communicate, do not disturb, away, invisible, extended-away, and offline) as commonly found in instant messaging systems, contact lists, user interface preferences, training application and training session preferences.

3.1.2 Lobby

The lobby in DARWorld couples many of the features found in web portals with capabilities often seen in game lobbies. The lobby is a browser-based environment that creates a private space to organize and provide ready access to various DARWorld information, such as training offerings, news, documents, calendars, and schedules; very much like a web portal. The lobby also provides mechanisms for managing that information and personalizing user preferences. The lobby provides access to shared calendars, schedules, shared composition repositories, such as Wikis, discussion forums, and collaboration tools like instant messaging systems. The lobby provides access to a variety of matchmaking and search capabilities, described in Section 3.1.4, which provide mechanisms for selecting and locating learning offerings, training systems, other users in DARWorld. Like a typical game lobby, the DARWorld lobby also provides access to collaboration services (see Section 3.1.3) for organizing team training exercises and creating, launching, and controlling the training session. Training session control capabilities and preferences include the ability for session administrators to control access to the training session (both prior to and during the session) start and end training sessions, request an out-of-session communication channel to members of the training session, control out-of-session communication mechanisms available to training session members, etc. Members of a training session can also set preferences for out-of-training incoming events such as news and other communication mediums such as chat, IM, or voice.

Some typical uses of the Lobby include:

- Scheduling training sessions and sharing calendars with trainees, instructors, observers, etc.
- Organizing and launching individual and team training exercises.
- Conducting discussions on everything from AARs to content authoring.
- Creating and sharing access controlled composition repositories such as Wikis.
- Managing and setting up team training exercises.
- Maintaining contact lists of instructors, trainees, team members, subject matter experts, etc.
- Posting news and announcements among the training community.
- Creating personalized learning plans using DARWorld curriculum components.

- Creating and maintaining a personal Profile Page.

3.1.3 Collaboration Environments

DARWorld will provide a number of collaborative tools for communication, information sharing, and to some degree, co-development. These collaboration tools will be invoked from the lobby and are tightly integrated with the lobby environment. Other applications and training systems may rely on the ability of users to utilize these collaborative tools.

There are several collaboration environments that will be part of DARWorld Command. Those collaborative tools are listed in the subsections below.

3.1.3.1 Instant Messaging

Instant Messaging provides an interactive environment for users to converse by typing short messages and responses (much like a regular conversation). The instant messaging capability is an important tool for the lobby environment (especially the exercise planning stage). With instant messaging capability, users will be able to hold conversations with other participants to organize individual and team training missions, to communicate to other members as part of the training mission, to assist during training, to communicate outside of a training exercise, and for many other peer to peer communication and collaboration needs.

3.1.3.2 Voice over IP

Voice over IP allows real-time voice communication through the DARWorld environment. This is especially useful for timing critical collaborative events and for times where hearing voice is important for training (e.g. language trainers, response to sense of urgency, etc.).

3.1.3.3 Email

DARWorld will provide a standard electronic mail server that allows standard email clients to be used. DARWorld users will have a DARWorld specific email account to which DARWorld email traffic is directed. The user may choose to forward this email to another account or use the DARWorld account directly. This collaboration means allows users to exchange messages when time criticality is not an issue. For example, email could be used in DARWorld for training assignments, and feedback and suggestions.

3.1.3.4 Wiki

A Wiki is a collection of web pages that can be edited by anyone, at any time, from anywhere. The main goal of the Wiki is to provide a mechanism for displaying and modifying shared information in an organized manner. The DARWorld Wiki will be integrated into the user database; it will use the user identities in the DARWorld database. Unlike some of the other collaboration environments, the Wiki is set up to share information among a larger audience (more than two or three collaborators). It will likely be used in DARWorld for feedback and suggestions, lists of frequently asked questions, and notes about prior events (briefings, news, etc.).

3.1.3.5 Additional Collaboration Tools

Additional collaboration environments will be integrated in DARWorld as needed. Potential DARWorld collaboration environments include whiteboards, forums, and blogs.

3.1.4 Matching Services

The matching services are much like a customized search engine to find the components (and the location of the components) required to begin the right training environment. It will be invoked from the lobby and is tightly integrated with the lobby environment. The matching services are focused on the following components:

- Training Packages for specific training needs. The training package search capability can incorporate profile information (e.g. level of expertise, past training experiences, etc.) to simplify the search parameters. For example, it will be possible for a user to search available training packages for one or more of the current training objectives with an option of excluding those for which the user is already competent. The result of the search will be a list of training package web pages specific to the search requirements.

- Team members to fill certain roles (semantic match making). If the chosen training scenario requires multiple participants, the other roles will need to be filled to execute the scenario. Locating additional participants (who they are and where they are) can take several forms depending on the time scale. The matching service addresses the short time scale and will provide a list of available participants that fill the requirements of the role. Additional participants could be others who are in the same situation, but with complementary needs. Other DARWorld participants can assert their willingness to participate in any session for which they are competent (or need training). On longer time scales, potential participants could receive email messages about events of potential interest to them or a user could search for upcoming events of interest to them.

3.1.5 Launching Service

After the lobby environment and tools are used to select a Training Package and participants, the launching service is used to launch the training systems . The launching service uses the Training Package and other parameters (players and IP addresses, etc.) to determine the bindings for each training system (or other application) in the package to ensure a correct startup process. The binding information (startup parameters, environment variables, etc.) is retrieved from training package descriptions, profiles, etc. The launching service also assists in the coordination and synchronization of the launching of multiple training systems (for team training). This launching capability helps minimize the number of miscues in starting, and allows the lobby to be the central control environment for DARWorld.

3.2 DARWorld Backend

3.2.1 Content Management Services

3.2.1.1 Content management

Content Storage Services

Content storage services provide a number of typical content management services, including version control, content consistency enforcement, check-in/check-out locking and control, and history and reporting capabilities. Although the content repositories are distributed in nature, storage services provide a layer that makes the distribution transparent to the content provider.

Registry Services

The registry stores descriptive and structural information (metadata) about available content data, as well as metadata about applications such as training systems.

Publishing Services

When new content is offered for release to the DARWorld community, it must be published and made available to the community. Publishing services manages this process, assigning release version numbers, and updating appropriate repositories with the content and metadata regarding the content. This metadata includes information regarding what and how a training system(s) can deploy this offering.

3.2.1.2 Training Package Administration & Management

Training package administration and management functions include those activities required to make training packages available to DARWorld users. This includes managing metadata descriptions and relationships to user goals and competencies, managing package versioning, and relationships between packages and training system capabilities. Also included is the management of resources available for delivering learning content, such as tracking system requirements for packages, defining resources for packages, adjusting resources based on system requirements, etc.

3.2.1.3 Training System Profiles & Management

Training system profiles are designed to describe the capabilities of a training system, enabling training package management services to properly bring together training selections, the training system and the trainee. The information in the training system profile must include enough information so that the interface application for the training system can be instantiated. This may include the resource requirements for the training system, its server support requirements, operating system compatibilities, etc.

3.2.2 Member Management Services

3.2.2.1 Member Profiles

Member profiles include personal information about a DARWorld member, including training objectives and plans, user schedules, contact information, and preferences regarding DARWorld applications including collaboration services, training system, and DARWorld client components. The member manages these data.

3.2.2.2 Team Profiles

Team profiles are an extension of member profiles that are designed to enable the organization of individuals into teams; thus, a principle extension of a team is a list of members. Teams may themselves be composed of other teams. Team profiles have privileged members, such as team leaders or supervisors, who may manage team information such as the team's training objectives and contact information.

3.2.2.3 Tracking Services

Tracking services track a trainee's progress through a DARWorld curriculum by recording the history, current status, and anticipated future progress through the training offerings. Trainees can modify their own learning development plan, but cannot modify system-based or instructor prescriptions. Privileged users such as instructors or superiors can access a trainee's learning plan and training history.

3.3 DARWorld Applications

3.3.1 Training Systems

The training system (client) is the central component for training in DARWorld. It is typically a standalone application that is launched from the lobby. The training system client is not as tightly integrated with the lobby environment as the other services (launching service, matching services, and collaboration services), but may rely on the existence of those services rather than provide duplicate functionality. For more information on the training system, refer to Section 5.

3.3.2 Content Authoring

We anticipate that content can be created in many ways, by many different parties – for example professional content developers creating specialized educational or training material; trainees posting assignments, modifying scenarios, self-critiquing performance, and adding commentary regarding existing content; instructors and experts annotating and augmenting after action review materials; or DARWorld librarians organizing content and authoring meta-data.

Content authoring may consist of:

- Adding new content to a collection.
- Reusing existing content in new situations or to address other training objectives.
- Modifying or annotating existing content.
- Adding or modifying meta-data about existing content.
- Evaluating content.

3.3.3 Database Import/Export

There is no direct support in the infrastructure for connecting DARWorld to foreign databases. Indeed such support could not be designed without a specification of the foreign database and the kinds of import or export. The easiest, and probably most common case is to import a database for the purpose of automating the registration authorization process. This process will require tailoring DARWorld Command to offer the option of registering in DARWorld using the information from the foreign database. A user choosing this option would supply certain information that a purpose-built registration authority could use to validate the user's registration information against the foreign database. In some cases the data in the foreign database would be mirrored in the DARWorld database, but this is not required and should be avoided if possible.

The second most common case is for the training requirements (objectives) of the personnel of an organization to be imported from a foreign database. This would again require specific software to translate the information of the foreign database into DARWorld and, in some cases, new training objectives would have to be designed and training packages built to match those objectives. On the other side, it may be desired to export the training outcomes back to the foreign database. Again, specific software would have to be built.

None of these external database activities require additional support from the DARWorld infrastructure; they build on the infrastructure already defined.

3.3.4 Other Tools

Other tools include management and visualization tools, e.g., applications that allow getting the system administrators view of DARWorld; applications to parameterize DARWorld client components; or applications to manage training packages, curriculums, or trainees.

4 Distributed DARWorld Infrastructure

This chapter describes the DARWorld architecture from a distributed component view. The chapter will explain what the different servers/daemons/clients/peers are and how they are interconnected.

4.1 Overview

As shown in Figure 4, there are two major pieces of the DARWorld infrastructure: the DARWorld Backend and various DARWorld Applications that connect as clients to the DARWorld Backend, interacting via a set of network protocols. This section goes through each of the modules of these two architectural pieces (see Figure 5) and describes the technologies and protocols used to create the distributed DARWorld infrastructure.

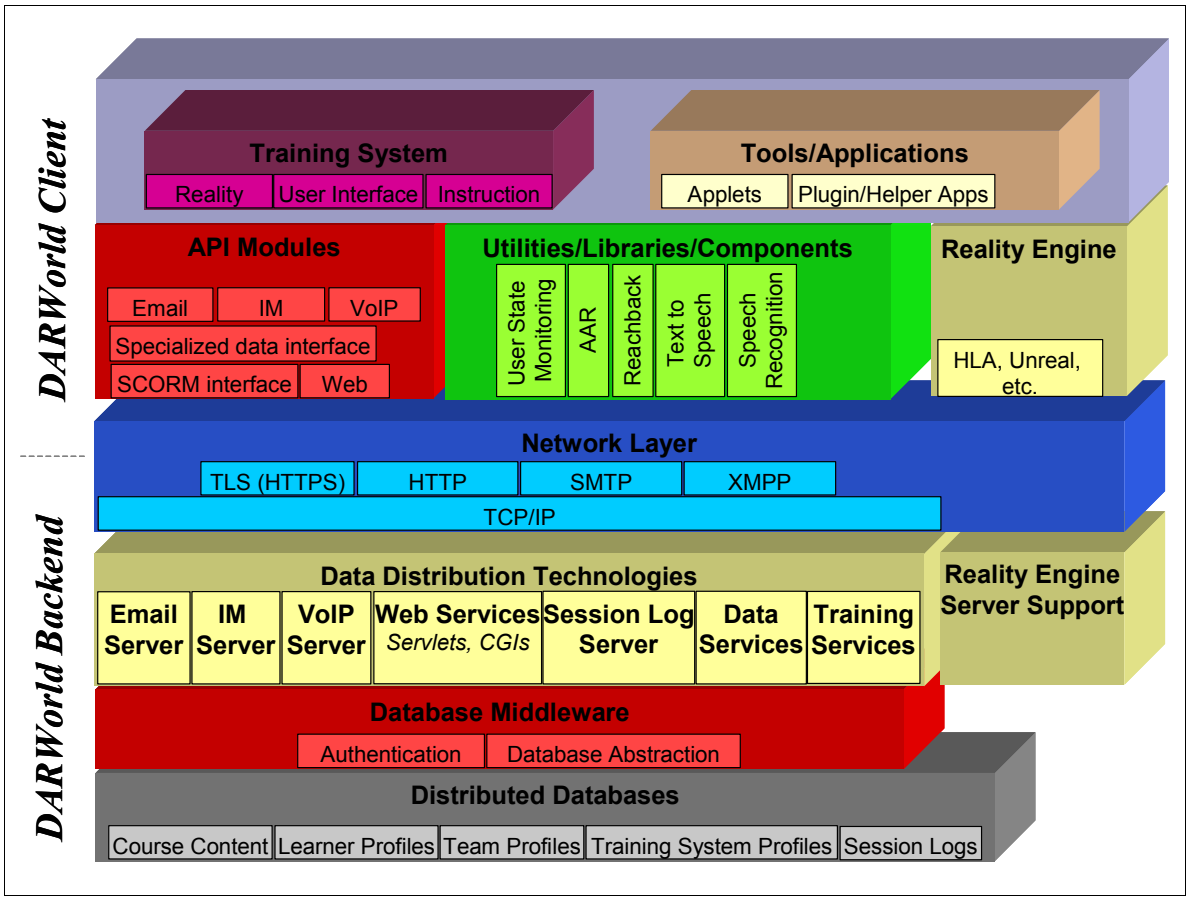

Figure 5: DARWorld Backend Modules and DARWorld Client Modules

While some modules can be used in multiple places, the following subsections follow the format of the figure and are thus divided into two subsections: Backend Modules (Section 4.2) and Client Modules (Section 4.3). They describe the individual modules and specifically discuss the purpose of the module, the interfaces, high level implementation information and security implication (if necessary).

4.2 DARWorld Backend Modules

4.2.1 Database

The database module is the central storage for all information that needs to be accessible to multiple applications and training systems. This (relational) database may be distributed across multiple platforms for scalability and reliability reasons. The types of data stored in this database can include:

- Training Content – training scenarios, SCORM content, etc.
- Member Profiles – scores, training objectives, preferences, identification information
- Team Profiles – scores, learner members and roles, training objectives
- Training System Profiles – application initialization parameters, application preferences.
- Session Logs – storage of events that occurred in any or all applications running in a session. This session information is tagged and time stamped for ease in data recovery
- Training Packages – used for the training session, includes locators for scenario files and descriptions of resources (e.g. servers) required.

There are many infrastructure services that use the database information. However, all interactions with the database are done using the container-managed persistence features of the J2EE container.

There are multiple COTS implementations of databases that will work well for this module. J2EE containers or database products provide resource adapters for the databases we might encounter. Standard database security is provided in this module to restrict database access to the J2EE container. The J2EE container provides fine-grained authorization and authentication mechanisms. See Section 8.5 for a description of the DARWorld security services.

4.2.2 Session Log Service

A session is defined as a training activity – from launching the training system instances through the start of the after action review. Throughout a session, both DARWorld Backend and Client events are logged and stored into the session log on the DARWorld Backend. Hence, the session log is a significant part of the database in size and activity. The session log service will support two major functions. First it will interface with a client (e.g. AAR) to transfer large amounts of data efficiently. Second, it will mediate multiple users and manage the information in the session log. All access to the session log (for both getting and setting) goes through the session log service.

Session log management includes:

- Providing information about what session log a user is associated with (a user can only be associated with a single session at a time).
- Placing the passed data in the correct session log (there will likely be many active session logs).

The implementation of a protocol to efficiently transfer the session log information to a client will likely be a COTS package. The implementation of the session log management will be software implemented specifically for DARWorld. The session log service interfaces to the SessionLog entities of the database and will likely be called by most of the other services (VoIP Server, IM Server, Email Server, Web Services, etc.).

4.2.3 Instant Messaging Server

The IM server running on the DARWorld Backend handles all DARWorld related instant messaging communication. All DARWorld related IM information is relayed through this server and is added to the session log (with tags and time-stamps). This information could be useful in determining the timing/reasoning of events in the AAR.

For logging purposes, it is required that the users that are communicating via IM be part of the same session. If the users are not part of the same session (e.g. during the initial collaboration stages and locator services), IM data will still be delivered, but will not be logged in the session log.

The implementation of this module is the Jabberd IM daemon. Modifications to the 3rd party software will be necessary to interface to the session log server to log the IM data to the session log.

This module interacts with Jabberd clients (running on the Client) through XMPP. Also, for logging, the module will be interfacing to the session log server module.

Data security will be supported in this module if provided by the 3rd party software and if security policy suggests that certain information transmitted over this protocol should be secured.

4.2.4 Email Server

The email server running on the DARWorld Backend handles all DARWorld related email traffic. All email correspondence is relayed through this server and is added to the session log (with tags and time-stamps). These email messages could be useful in determining the timing of events in the AAR.

For logging purposes, it is required that the email sender and receiver be part of the same session. If not (e.g. during the initial collaboration stages and locator services), email communication will still happen, but will not be logged in the session log.

The implementation of this module is a standard COTS SMTP server. With a standard SMTP server, the client software on the client can have a standard email interface configured to the DARWorld SMTP server. Modifications to the 3rd party software will be necessary to log the relayed email traffic to the session log.

The email server will interface with the session log server module to log the relayed email messages into the database. Data security will be supported in this module if provided by the 3rd party software and if security policy suggests that certain information transmitted over this protocol should be secured.

4.2.5 VoIP Server

The Voice over IP (VoIP) server provides the server support to set up peer-to-peer and conference connections within the DARWorld system. Initially, voice data transmitted is not logged on the server. The only information that will be recorded in the session log is the connections that are set up between the participants.

The implementation of this module is purpose-built software. This module interfaces with the VoIP client software on the client through the VoIP protocols.

4.2.6 Web-based Services

The web-based services module is the principal remote interface to the database. With the help of servlets and CGIs, information or requests for information can be transferred from the client or training system to the database. Using a web server provides a standard interface for information in many different forms to be organized, stored and retrieved from the database. Some examples of how web-based services could be used within the DARWorld system are as follows:

- Registration – passing the identification information and retrieving the profile information
- Lobby – gathering profile information about the next training session and requesting content for specific training.
- Whiteboards – collaboration efforts.

The web-based services module handles the incoming data and security of the data; management of that data is done through servlets and CGIs. The web server interfaces to the servlets and CGIs as well as the network layer interface to web clients are all standard. Interactions to the database are done through the servlets and CGIs.

Implementation will likely be the latest version of Apache[Apache]. There are a couple of COTS SSL extensions to Apache that could be used to provide a secure connection between client and server (e.g. Apache-SSL[Ap.SSL] or mod_ssl[modssl]. It is assumed that most information transmitted through the web services channel will be secured. Refer to the security Section 8.5 for more details about the security provided. No modifications to the 3rd party software (outside of the SSL extensions) are expected.

4.2.7 Servlets

The purpose of servlets within the DARWorld system is to manage incoming data from the web server. The servlets will receive and parse the information from the web server and as a result will interact with the database. A significant part of the server access in DARWorld is through Servlet.

If possible, off-the-shelf software will be used, but there are no obvious choices now and it is likely that much of the servlet software will need to be written specifically for DARWorld. One servlet that is likely to be COTS is the SCORM servlet. Listed below are some of the servlets that are provided / will be needed in DARWorld:

- Registration – parse the passed registration information and create a profile in the database
- Locator – parse the locator request and search through the database to find the desired information.
- Lobby – parse the request, set preferences, get training offerings, etc.
- SCORM servlet – COTS package to manage web-based training information

The web server controls supplies the servlets with the certificate (identity) of the user. The servlets and CGIs are responsible for determining if the user is allowed to access the information they manipulate.

4.2.8 Training Services

The training services module contains the following pieces:

- Package Administration Service – manages resources available for delivering learning content, such as tracking system requirements for packages, assigning resources to packages, adjusting resources based on system requirements, etc.
- Training Package Management Service - makes packages available to DARWorld users. This includes managing package metadata descriptions and relationships to user goals and competencies, managing package versioning, and relationships between packages and training system capabilities.

- Member Profile Service – manages member and team profile information including training objectives and plans, team members, user schedules, contact information, and preferences regarding DARWorld applications including collaboration services, training system, and client interface components. The member manages these data.

- Tracking Service - tracks a trainee's progress through a DARWorld curriculum by recording the history, current status, and anticipated future progress through the training offerings.

Refer to Figure 4 and Section 3.2 for a more detailed description of the training services.

4.2.9 Reality Engine Server Support

The Reality Engine Server Support module provides the server support necessary to run a reality engine. For example, an HLA federation may require a federation execution process that would run on a server. DARWorld supports multiple reality engines and the implementation is different depending on the reality engine used. The need for a reality server support is declared by the training system profile and inherited by the Training Package.

The Training Package will specify the bindings (executable name and arguments) for the reality engine server component. There will be a mechanism that manages reality servers and provides information about the available reality servers to the launcher (which would include the reality server in the bindings). The launcher is responsible for making sure an available launching server is running (if necessary).

The interface to this module is dependent on the reality engine protocols and structure and is not defined by DARWorld. Data security is provided only if the reality engine server supports it.

4.3 DARWorld Client Modules

4.3.1 API modules

API modules provide interfaces and client software to communicate with the DARWorld Backend modules (e.g. IM Server, Email Server, Web Services, etc.). These client software modules and APIs are described in more detail in Section 6.

The training system, the tools and applications and the shared utilities, libraries and components all call the API and client software modules. In turn the client software modules interface with network layer protocols to send/receive information with other clients and the DARWorld Backend.

Implementations of many of these modules will be based on standards and COTS and are described in Section 6.

4.3.2 Utilities/Libraries/Components

Utilities, Libraries and Components are a collection of reusable modules for help in developing a training system and an application/tool. These commonly used modules will help reduce development cost for new training system efforts and help reduce integration time and cost associated with integrating an existing training system. This collection will likely grow as new modules are created and added. Some of the modules that are included in this collection would be:

- User State Monitoring Library

- AAR Component

- Reachback Component

- Text to Speech Utility

- Speech Recognition Library
- Instant Messaging Component
- Observer Component

These utilities, libraries and components can interface with the API Modules, the Reality Engine Interface (if there is a module that is specific for a particular reality engine) and the network layer. The training system and Tools/Applications modules would be the callers of these modules.

Implementations of these modules will range from COTS packages (e.g. speech recognition libraries, text to speech utility, etc.) to specifically developed software for DARWorld (e.g. user state monitoring library).

4.3.3 Reality Engine

The Reality Engine carries the underlying simulation/reality information between training system applications (and potentially within an training system component). DARWorld will support several different simulation/reality systems and protocols as necessary to support the execution of both legacy and evolving training systems. DARWorld will not develop Reality Engines, but DARWorld will contain COTS packages for a subset of the Reality Engines (e.g. certain HLA federates, Unreal). If possible and necessary, there will be modifications to the COTS package to provide DARWorld specific extensions and enhancements (e.g. enhancements for critical logging support, etc.).

The Reality Engine interfaces to the networking protocols and is called by the training system, the Tools and Applications, and potentially some modules in Utilities/Libraries/Components. Supporting security in this subsystem is dependent on the implementation of the simulation/reality systems.

4.3.4 Tools/Applications

A DARWorld application or tool is an individual application built on top of the DARWorld infrastructure. The implementations of tools and applications will vary based on the purpose. It is unlikely that COTS will fill the need for most applications. The applications drive the security requirements. If security is required, it will use the services that support the appropriate level of security.

Toolsand applications can interface to the API modules, the Utilities/Libraries/Components and potentially the Reality Engine. Applications can use few or many of the provided services, at their option. A training system that supports more APIs provides more functionality to the end-user. As training system software matures, it is expected that it will significantly increase its integration with the system.

Three basic approaches, to building the tools and applications, are listed below.

4.3.4.1 Applets

Applets are an application that runs within a browser. Applets interface with the API modules the Utilities/Libraries/Components and potentially the Reality Engine. An example of an applet within DARWorld is the SCORM run-time environment presentation applet.

Typically applets are used for tasks that can be accomplished within a browser environment.

4.3.4.2 Plugins/Helper Apps

Plugins and Helper Applications allow applications to interface with browser-based system. The purpose of this is to support seamless navigation from browser-based interactions into standalone applications. Many plugins and helper apps will be COTS, but some will require new software development. Security is supported depending on the plugin or helper app.

4.3.4.3 Standalone Executables

Typically, standalone executable applications are purpose built for cases where a browser/applet approach is unsuitable. An example might be a Session Descriptor Editor or even a training system.

4.3.5 Training System

A training system can be considered a standalone executable application. Like the other applications, it uses as many of the provided services as desired and interfaces with the API modules, the Utilities/Libraries/Components and the Reality Engine. Refer to Section 5 for a more detailed description of how a training system fits in the DARWorld environment.

4.4 Network Layer

Both the DARWorld Backend modules and the DARWorld Client modules interface to a network layer. This network layer is made up of COTS packages for networking and transport services. Those services include:

HTTP – the standard web based protocol – no data security

TLS (HTTPS) – the secure version of HTTP. It supports authentication of the client and server as well as ensuring data privacy and integrity.

SMTP – Simple Mail Transfer Protocol – standard underlying email service.

XMPP – Extensible Messaging and Presence Protocol – used for instant messaging data

TCP/IP – Underlying transport mechanism used by the above protocols.

5 Generic Training System Architecture

The DARWorld architecture is designed to facilitate the development of training systems, provide mechanisms for the delivery of and integration of those training systems into the larger training curriculum, and add to the utility of new and existing training systems by enabling them to function within larger team training exercises, or as components enhancing other individual training systems. There are a number of ways in which DARWorld, and the DARWorld architecture can facilitate training system development. The first is by reducing the overhead in connecting a preexisting or new training system to DARWorld Backend systems by providing a set of key services, components, libraries, and protocols that enable interoperability. The second is by making available to the training system developer community those services, components, or libraries that are useful to more than one training system developer. In some circumstances, these tools may exist in a form that most training system developers can reuse without modification. In other cases, the architecture developers may be required to adapt or wrap these components to make them more useable by the development community at large. Finally, DARWorld facilitates training system development through the specification of a high-level training system architecture that provides a framework by which training system might be designed and understood, and promotes interoperability through the identification of key components and interfaces.

The IEEE Learning Technology Standards Committee has developed the Learning Technology Systems Architecture (LTSA) which specifies a high-level architecture for information technology-supported learning, education, and training systems (for details see Section 11.1). The LTSA specification was designed to be generic enough to encompass virtually any training system, and is independent of any implementation details such as the actual APIs and protocols used. It is our belief that this level of abstraction is the right approach to describing what it means to be a training system in DARWorld, and for training system components to interoperate and to be reused within DARWorld. The DARWorld architecture, with its intent to facilitate simulation-based team training capabilities emphasizes a number of processes, information repositories, and information interfaces not highlighted in the LTSA. These are elaborated below.

5.1 Training System Interoperability

We believe there is value in facilitating the interoperation of training systems and training system components in a number of ways. Training system interoperation obviously depends on the types of training systems offer, on the maturity of a training system, and on training system structure.The following section describes different facets of training system interoperability.

First is the use of instructional components, not originally designed for that training system, by a training system. This might include instructional components originally developed for one training system and reused by another training system. For example, a coaching module that was originally designed for one training system might contain instructional and domain expertise that is relevant to another training system. This would promote its reuse. This is illustrated in Panel A of Figure 6 in which a primary training system uses the instructional component from a secondary training system. Alternatively, instructional components could be designed as third-party components specifically for inclusion by training system developers. For example, a team training evaluation component might be generically designed to evaluate team cohesiveness across training systems. Instructional components reused in another training system will require access to the behaviors and actions of the user as well as the content to which the user is reacting, including any simulated reality in which the user is immersed. Furthermore, the instructional component must be able to interact with the user, mediated in such a way as to not conflict with other instructional components. The instructional component may, in some cases, need to manipulate the reality simulation in order to promote, or facilitate instruction. The level of interoperability required by the instructional component will in many cases vary greatly. The degree to which adaptors or moderators (i.e., translation components) will be required in order to provide interoperability will also vary on a case-by-case basis.

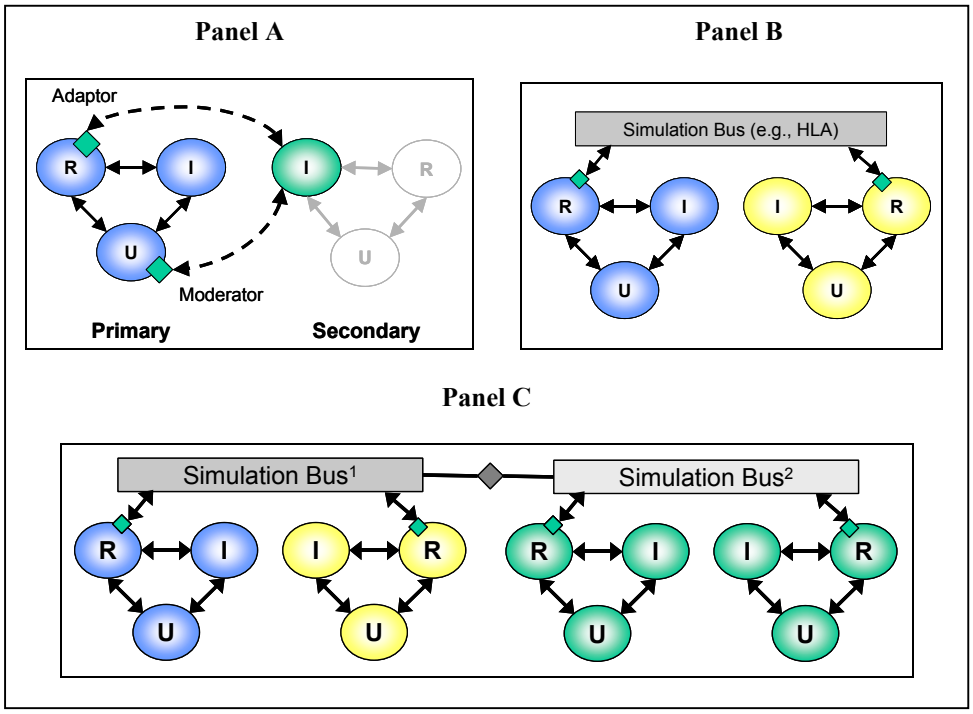

Figure 6: Types of training system Interoperation

A second method of training system interoperability occurs through simulation interoperability. Figure 6, panels B and C illustrate two different classes of simulation interoperability. Panel B illustrates two training systems interoperating through a shared reality substrate. Training systems not originally designed for interoperation, but using a compatible simulation bus (such as HLA) will likely require adaptation to ensure a similar event language. Additionally, training systems that do not share a realty substrate like HLA, but instead use different reality generators, where one training system uses a reality substrate such as HLA and another uses another reality generator, (e.g., the Unreal Engine[3]) additional work will be required to develop the adaptors necessary to enable interoperability. We anticipate that as these adaptations occur, the DARWorld architecture team will promote their reuse by other DARWARS developers. Lastly, panel C, depicts two reality substrates interfaced directly through an adaptor.

5.2 Training System Component Interfaces

It is our belief that the DARWARS community will develop, over time, a number of components that will be useful to a large number of training system developers. We anticipate that many of these components will be developed by the DARWARS architecture team themselves, and will include libraries and services previously described in Sections 3 and 4 and used to help a training system operate within the larger DARWorld system. Additionally, training system developers will construct components that will be useful to other training system developers. It is our expectation that the components most easily adapted and reused, will beparts of user interface component, rather than the instructional or reality generation component, of a training system. Figure 7 illustrates an idealized user interface component made up of subcomponents or services we anticipate becoming available over the course of the DARWARS program to other training system developers.

[3] http://udn.epicgames.com/

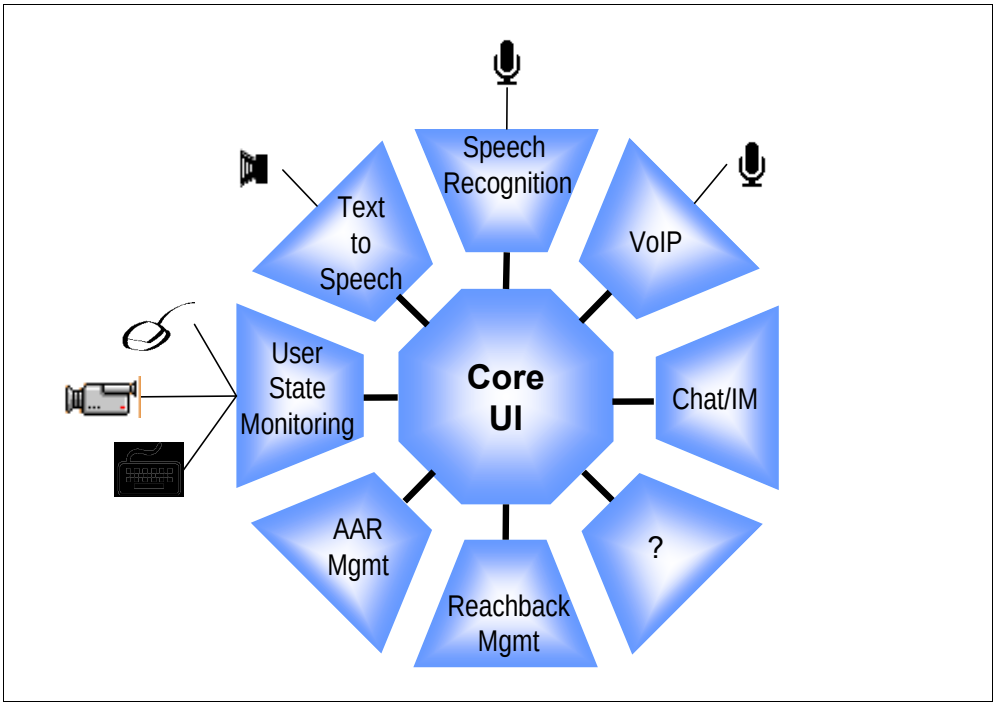

Figure 7: Subcomponent View of the User Interface Component of a training system

The component developers themselves will specify the interfaces (code, APIs, protocols) of these components. We do not attempt to specify what these interfaces must be in order to be useful as a DARWorld component. It is, however, critical that those interfaces be made publicly available to the DARWARS community, and it may be advantageous to the DARWARS community that those components be augmented or adapted to better suit the DARWARS community at large. The DARWARS architecture team should play a role in making those components available, as best serves the DARWARS community.

6 Infrastructure Interfaces

6.1 Overview

The services provided by the components of chapter 4 are accessed using specified protocols. Applications that use those services must send and receive messages according to those protocols. There must be software to send and receive messages according to the protocols. For applications written in the Java programming language, the DARWorld infrastructure will supply a library of protocol modules, which implement those protocols and provide convenient access methods. Applications written in other languages will need to acquire or implement comparable software written in those other languages. In the case of C or C++, the implementation could be achieved by including a Java Virtual Machine (JVM) and use the Java library components through the Java Native Interface[JNI] (JNI). The Java implementation should be considered a reference for implementations in other languages.

This chapter mainly describes the facilities available to the developer in terms of the components that provide interfaces to backend services, but where necessary, the important protocol information will be included. Most of the protocols will be based on SOAP[SOAP]. In most cases, the SOAP messages will use the HTTPS[HTTPS] transport. In cases where a web server does not furnish the service implementation, the transport protocol will in most cases be BEEP[BEEP]. In rare cases, protocols designed specifically for DARWorld will be used to enhance performance. The protocol details will be specified as part of the design process and only the generalizations about these protocols and components will be described here.

There is no single DARWorld application architecture. However, uniform access to the distributed services will be encouraged by supplying a Java library of components to perform the access and by establishing a clearinghouse for development of interface components in other programming languages.

The components described here are mainly for use by tools and training systems built on the DARWorld infrastructure. But, they will also be used by some of the components of the distributed infrastructure itself. Furthermore, the peer-to-peer protocols between, for example, the components of a training system are much like those between client and server. Some of these components are included in this chapter, though strictly speaking, they are not “infrastructure” components.

The services that are available through these interfaces may place restrictions on what features are actually allowed according to the identity of the user. The components described here are not responsible for access control. They are responsible for establishing and maintaining a user identity (principal) and furnishing that through the secure connection to the server. Security issues are addressed in more detail in Section 8.5, but for now it is sufficient to know that client certificates serve the basis for access control.

6.2 The Typical DARWorld Application

Figure 8 shows the layering of a typical DARWorld application. A layer of DARWorld support components operates between the application and the network transport to provide a procedural interface to the protocols used to access the distributed services of the DARWorld Backend. It also shows the distributed service implementations in the backend to which these API components connect to perform their tasks. It is not necessary for every application to use all of these interfaces.

The diagram shows the three-circle representation of a training system. Only training system applications would have these elements and the components they represent. The interfaces within the training system are not detailed in this figure, but are described below in Section 6.3 and have been motivated in Chapter 5.

The components that provide access to the DARWorld Backend as well as the utility components have a range of relevance for instruction. The diagram separates certain components as “Instruction” components. While the rest of the interfaces have some instructional relevance (remember the blue-book instructions to print your name on the cover), the highlighted interfaces provide the access need to perform the learning management functions of DARWorld.

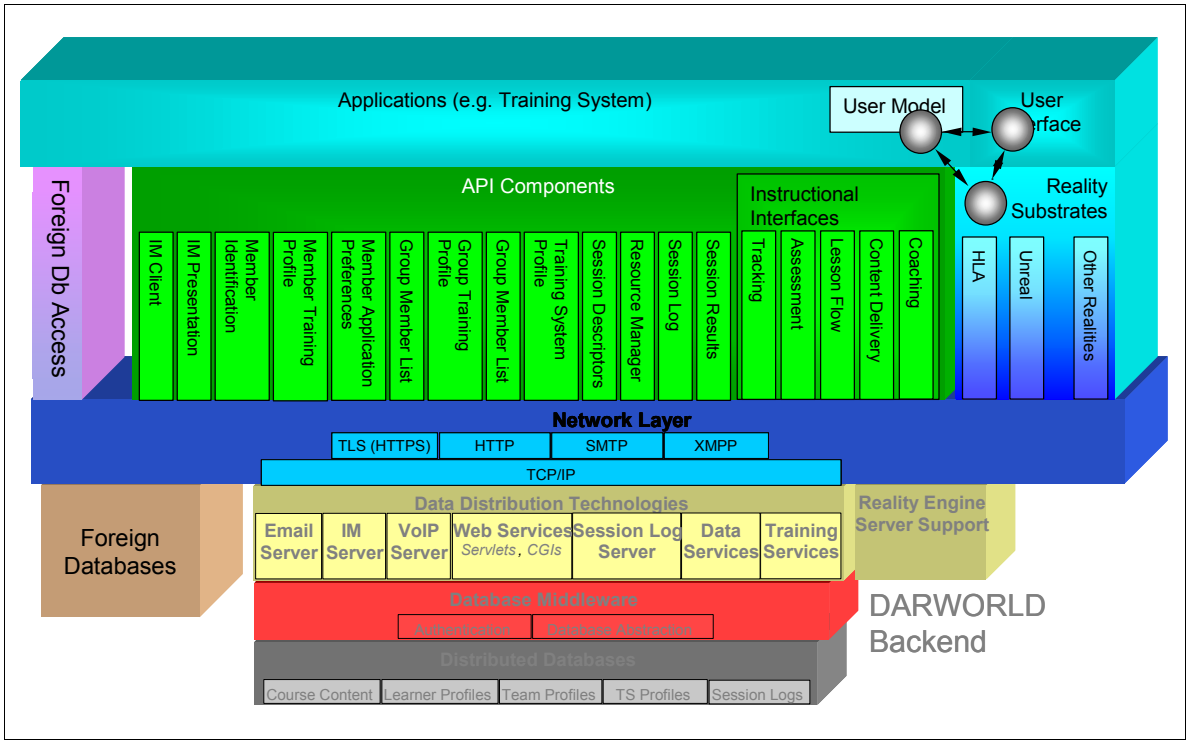

Figure 8: Layering of a typical DARWorld Application

6.3 Intra-Training System Interfaces

Chapter 5 discusses the interaction between three sets of components within a training system. Here we describe the components that enable those interactions. An intelligent tutor has the job of trying to develop insight into what a student is doing and to judge if or what advice or comments should be offered. The bases for this insight are what is happening in the (simulated) world and what the user is doing. The latter is sometimes called user state modeling and can be both passive – observing his use of controls, gaze tracking, physiological measurements, etc. and active – asking questions.

The user state model is logically part of the instructional component of the system though the raw information for determining user state comes from the user interface. Some of this information consists of nothing more than observing the actions of the user as he uses the controls of the interface. Other information comes from hardware and software, e.g. a camera, specifically inserted into the user interface to gather information to help develop the user model.

Intelligent tutor interfaces are those interfaces between the instructional (I), reality (R), and user interface (U) parts of the training system (Figure 9). They differ from the other interfaces of this section in that the interactions are between client-side components (peer-to-peer) rather than between client and server[4]. However, if the components execute in different processes, the form of the communications is similar to that of the other interfaces here. The protocol will be SOAP using BEEP transport. HTTP transport is inappropriate because no web server is involved. Because these interfaces are likely to be highly specialized and unique, DARWorld does not attempt to dictate their specifics, but the infrastructure protocols can serve as models for the intra-training system interfaces.

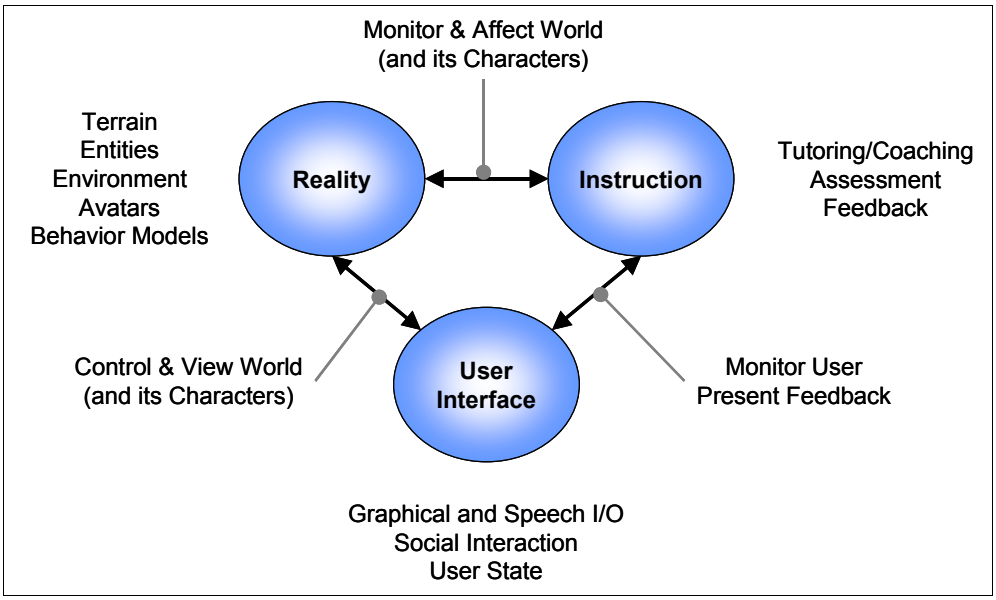

Figure 9: Intelligent Tutor Interfaces within training systems (cf. Figure 1.)

6.3.1 Instruction – Reality Engine

Sensors installed in or connected to the reality substrate are used to determine what is going on in the world. Sensors that simply connect to the reality substrate are straightforward and use the advertised features of the substrate. In cases where the desired information is not available from the standard API of the reality substrate that API must be modified and sensors installed within the substrate to sense the needed information. The feasibility of this depends strongly on the nature of the reality substrate because it probably requires access to the source code and that may be expensive to acquire and understand especially for proprietary reality substrates.

Both kinds of sensors detect pedagogically relevant events and feed those events through this interface to a tutor that interprets those events. The tutor may conclude that entering a note in the session log for AAR is sufficient or it may consider one or more direct interventions. The interventions chosen are based on the user's cognitive state (e.g. how busy) and the availability of suitable intervention modalities. For example, an AI in the simulation could be instructed to offer advice or ask a pointed question. When the intervention is to cause a non-human entity in the reality to perform an action, this interface must accommodate the control of that entity.

[4] The distinction between peer-peer and client-server interactions is frequently given undue weight. The only real distinction is which party initiates the interaction. Servers do not initiate interactions. Once an interaction has been initiated, the distinction disappears, though some protocols preclude an extended dialog.

Details of this interface cannot be specified at this time because they depend heavily on the API of reality substrate being used and the details of entities in that reality. Furthermore, every class of sensor is likely to have a different set of configuration parameters and will produce. Sensor setup specifies a set of attributes (attribute value pairs), a connection specification (how it is to be connected to the reality) and a reporting channel (which the sensor uses to report its events). Ideally, sensors can built as reality-independent components plus reality adaptors. The former contains the sensor logic, and the latter hooks the sensor into the reality.

Tutors can either be embedded in the application or accessed remotely. For reasons of performance, embedded software may be necessary and the interfaces would be defined in terms of their methods or procedures. But, remote tutors are possible and may be essential in the case of human tutors. For remote tutors, protocols must be defined as outlined above.

6.3.2 Instruction – User Interface

The interface between the instructional components and the UI components allows the instructional components to gather information about the user state and to affect the user's view to provide instructional feedback. This interface is really several interfaces depending on what user state sensors are available. For example, if a gaze tracker were available, then it would have an interface to establish its parameters and report tracking results. Similarly, the various GUI controls could be monitored for activity and each such control might have a different reporting capability. The same considerations of embedded versus remote components applies to user state components.

6.3.3 User Interface – Reality Engine

This is a highly idiosyncratic interface that has no particular instructional relevance. The user interface is almost always built to work with the chosen reality engine. This document makes no attempt to specify this interface.

6.4 Learning Management Interfaces

DARWorld has a very strong learning management aspect. Almost from the moment that a user logs in he could be dealing with learning management issues. Even a simple act such as instant messaging one of his buddies might be part of the process of gathering a group of individuals to start a training session. Certainly, by the time a user is running within a training system, he is well inside the learning management system.

A training system is an application that could incorporate many of the client-side parts of the DARWorld LMS. In particular, Content Delivery could deliver content to be shown to the user (scenarios, pre-briefs, etc.), Lesson Flow could decide what material should next be delivered based on the updates to the students training status that could be sent to the Tracking using data extracted from the student's performance as assessed by Assessment.

The following sections describe components and protocols that can be incorporated into a training system and implement the client side the learning management function. The incorporation of these features is optional, but is likely to be beneficial.

6.4.1 Assessment

The assessment and testing service evaluates performance during the training session or assesses learning with explicit tests. The assessment and testing process may involve the participation of instructors or other qualified personnel or may be automatic. Assessment modules are not infrastructure components, but the importance of their pedagogical role is sufficient to warrant mention.

Assessment components must understand the training that a training system performs. Assessment has two parts: measuring performance and relating the measurements to training objectives. The first cannot be generic; it must be training system specific. The second, however, is a generic process and infrastructure components will be available to do it.

Performance measures and the mapping of those measures from training objectives (cf. Figure 1) serve as inputs to the generic assessment process. The assessment produces a multi-dimensional measurement for each objective. The axes of the multi-dimensional measurement are labeled by the skill and situation/event between the objective and the measure.

DARWorld maintains the products of the assessment (see Section "Tracking" below).

6.4.2 Tracking

The tracking service stores the student's performance (training achievement) for future reference and for use by the sequencing service to determine what to do next. Tracking is done in terms of training objectives. The overall form of the objectives can be specified by SCORM (cf. Section 11.1.1.1). We expect training objectives to be developed by the organizations that use DARWorld. However, the DARWorld tracking service will record multi-dimensional measurements of performance against the objectives.

This component interacts with the member management services to update a student's records with a result associated with the training content just completed. This is roughly equivalent to putting a grade in a grade book. This information is ultimately stored in the student's training profile.

6.4.3 Lesson Flow

Lesson flow refers to the progression of instructional events in a training session. In the simplest case, the lesson flow is rigid and proceeds relentlessly from start to finish. More refined instructional components will tailor the instruction based on how they perceive the student's progress. The intelligent tutor of every training system is likely to have its own complex rules for deciding how the course of instruction should proceed. But, the various training system instances need a way to coordinate their sequencing decisions and need access to rules that control this decision-making process so that all the participants stay on the same page. Detailed definitions of these rules and coordination information await a better understanding of the kinds of flow needed and the kinds of coordination required. We expect these rules to be specified by the integration team with input from the training system developers.

The DARWorld lesson flow is more elaborate than the simple sequencing of SCORM 1.3, but it should devolve to simple sequencing for simple cases.

6.4.4 Content Delivery

The function of this interface is to deliver learning content to the application for display to the student. In many cases, the content is quite complex and consists of training packages, scenarios, and other forms of learning content and assets to specify complete simulation based lessons for multiple students. At the other end of the spectrum are simple web pages. For performance reasons, it may be necessary to add caching of the material being delivered in the form of a standard web proxy cache. The format of the content depends on its content-type

6.4.5 User Interface

The user interface of an application has the displays and controls relevant to the application's purpose. The user interface is specific to each training system and the DARWorld architecture does not assume or require any standardization.

6.4.6 Reality Substrate

There is no single DARWorld reality, but all training occurs within some representation of reality that the student observes and acts upon. Some representations are trivial such as web forms that he fills out. Others may be arbitrarily complex such as an HLA federation of multiple federates, servers, and other entities. The reality substrate for a training session is manifested in every training system instance involved as well as in servers designed to support that reality.

There is no single DARWorld reality substrate because each kind of reality substrate brings a set of strengths and weaknesses and each training system will choose the most suitable reality substrate based on those strengths and weaknesses.

6.5 Collaboration

6.5.1 Instant Messaging Client API

Instant messaging is a collaboration tool that will be used extensively in DARWorld to provide communication channels among its members. It is possible that instant messaging could be an important part of a training system either as an adjunct for obtaining help from subject matter experts, teammates, and instructors or as an integral part of the training. DARWorld Instant Messaging uses the XMPP (extensible messaging and presence protocol, a.k.a. Jabber) using SSL 3.0 (TLS 1.0) security as implemented by Jabber 2.0.

IM is a client-server protocol. The server maintains information about where the members are (presence) and distributes messages among them according to what connections have been established. IM client interface gives applications (e.g. training system instance) access to the IM server. This allows the IM function to be integrated with the other user interface displays. The Jabber client side is itself split into two parts: an API for communicating with the server and a UI for sending and receiving messages. The standard UI of an IM client would not, in general, be integrated with the rest of the application; it would simply co-exist on the same screen as the other UI artifacts. However, a training system developer can choose to provide his own instant message UI that is integrated into the rest of the UI. There are several XMPP client libraries available including JSO and YAJA (Java) and Jabberoo (C/C++).

A standalone IM application can also be used if the training system developer has no need for integrating instant messaging into his application. IM client interface gives applications (e.g. training system instance) access to the IM server. The IM server provides presence management and message exchange. This allows the IM function to be integrated with the other user interface displays.

A useful extension of instant messaging is to use it for controlling applications that normally do not interact with human users (e.g. services). Such applications can use the IM client component to receive commands from system administrators and to show status. Services using this feature would have a unique screen name and would be registered “members”.

6.5.2 Instant Messaging Presentation

Jabber clients include a UI for displaying messages, buddy lists, and accepting input. There are a number of available clients in various programming languages. Good Jabber clients separate the UI part from the API for accessing the server. Applications can avoid implementing their own IM UI by including the standard UI. This component, if used, is connected back-to-back with the IM interface module. Using this component is optional. A separate IM can always be used as long as the training system allows foreign windows to appear on the screen.

6.6 Member Data

These components provide interfaces to the member data services. Individuals and groups have much in common and the data structures take advantage of that fact. More detail about this data can be found later in Chapter 8. For the purposes of this section, the member data access is described as a number of sub-interfaces. Members have a unique membership number that is used throughout DARWorld to identify the member, but the number is seldom visible. The term "member" is used loosely. Most members are individual persons, but groups and services can be registered, too (6.5.1).

Some of the DARWorld services will be supplied by third-party software that requires persistent storage about the members it serves. This information must be integrated with the member data. This is particularly true for collaboration tools such as instant messaging. It is also true for the Wiki and other servers that need to associate the identity of a user with the other information they manage. Ideally, the server would use the member database for this so redundancy would be unnecessary. In the interim, the software will explicitly recognize and remove discrepancies in the redundant data.

6.6.1 Member Identification

Retrieves and updates member identification data. Identification data is information such as name, address, telephone, etc. The information available depends on what the database has available, which in turn depends on the overall system requirements. The database design will provide for variation in a user's identification over time. Identification information is accessed using the web services furnished by servlets in the web server. Access is by SOAP/HTTPS protocols and is controlled based on the identity of the accessor (cf. Section 8.5).

The key to most data is the membership unique id, but queries shall be available for finding the unique id given other information.

6.6.2 Member Training Profile

This component provides access to training-related information about a member. This interface is much the same as the one in Section 6.6.1 (Member Identification), except that it deals with training objectives and achievements. Training profile information is accessed using the web services furnished by servlets in the web server. Access is by SOAP/HTTPS protocols and is controlled based on the identity of the accessor (cf. Section 8.5).

There are three main uses of this component. A student or training supervisor may examine and update the student's training objectives. This activity is mainly done through web forms, but this API supports purpose-built learning management applications.

Training profile information may be searched (subject to access control) to find members that may be qualified to participate in and/or may benefit from participating in a training session. This ability can be used during the lobby activity to fill out the roster of participants.

Finally, a student's training objectives can be updated with the outcome of a training session. Tracking (Section 6.4.2) uses this to update student records. Training results are stored as multi-dimensional measures with axes specified by the objective and values determined by the assessment module(s).

6.6.3 Member Application Preferences

Most applications have a way to remember a user's preferences. Examples of preferences include window position, colors, fonts, and control preferences. This component allows the preferences to be stored centrally so it does not matter what machine a user is using. This is in contrast to the usual implementation where each application registers preferences on a single machine.

Preferences are stored as key-value pairs where the key includes the identity of the application and the value is a string. The standard implementation of this component will provide caching and lazy update to control network traffic and will provide common conversion facilities for numbers, Booleans, etc. Caching is possible because it is unlikely that a given user is using multiple instances of the same application and the consequences are not severe if he happens to do so. Lazy update batches updates for efficiency and to avoid multiple updates of the same preferences in rapid succession (e.g. while repositioning a window).

Access to user preferences is by SOAP/HTTPS protocols and is controlled based on the identity of the accessor (user). See Section 8.5. The data is stored in the central database and is mediated by a servlet of the web server.

6.7 Group (Team) Data

Group data is very similar to member data and is largely a subset thereof. For example, it is not likely for a group to have application-specific preferences. On the other hand groups must list their members. Groups may contain groups. Groups have a unique group number drawn from the same number space as member numbers that is used throughout DARWARS software to identify the group, but the number is seldom visible. Teams are a kind of group and the terms can be used interchangeably.

6.7.1 Group Identification

Retrieves and updates group identification data. Identification data is information such as name, address and telephone of group contact person, etc. The information available depends on what the database has available and is will be specified in the design stage. Identification information is accessed using the web services furnished by servlets in the web server. Access is by SOAP/HTTPS protocols and is controlled based on the identity of the accessor (cf. Section 8.5.)

The key to most data is the membership id number or screen name, but queries are available for finding the id number given other information.

6.7.2 Group Training Profile

This component provides access to training-related information about a group. This interface is much the same as 6.7.1, except it specifies the profile for a group. Training profile information is accessed using the web services furnished by servlets in the web server. Access is by SOAP/HTTPS protocols and is controlled based on the identity of the accessor (cf. Section 8.5).

There are three main uses of this component. A group leader or training supervisor may examine and update the group's training objectives. A group member may examine his group's training objectives and achievements. This activity is mainly done through web forms, but this API supports DARWorld specific learning management applications.

Training profile information may be searched (subject to access control) to find groups that may be qualified to participate in and/or may benefit from participating in a training session. This ability can be used during the lobby activity to fill out the roster of participants.

Finally, a group's training objectives can be updated with the outcome of a training session. Tracking (Section 6.4.2) uses this to update group records.

6.7.3 Group Member List

This component allows the list of members to be retrieved, searched, or updated. The members of a group can be determined, as can the groups of a member. Certain members of a group may be distinguished because they have access to modify certain group data, but this additional access is provide by the standard access control mechanism. There is no requirement that such access be restricted to group members.

The members of a group are specified in terms of the member unique ids. All members of a group have an access control role named by the group name. This allows the members of a group to access DARWorld items permitting access by the group.

6.8 Training Session

This section describes the components and protocols that are focused on a training session. They characterize the training system.

6.8.1 Training System Profile

A training system profile is a description of the capabilities of a training system – what it can do. The training system profile is a machine-readable specification of how to run the training system. The information in a training system Profile must be sufficient to create a user interface for a training package editor. Every option, parameter, and operating mode must be listed with explanatory descriptions and legal values. Every requirement for supporting servers must be spelled out. The training package editor allows an instructor to nail down specific options, parameters, and modes to achieve a training objective.

The training system profile does not refer to the training objectives that using the training system addresses, but it should provide guidance to help someone familiar with an organization's training objectives decide which training objectives would be addressed by the operation of a training system in a particular way. This guidance will require human intelligence to interpret.

6.8.2 Training Package

The final product of the lobby activity is the selection of a training package that is to be used for the training session. The training session setup component provides access to the training packages and the data they refer to, including locators for scenario files and descriptions of resources (e.g. servers) required. The selection of a training package involves queries for those compatible with the training objectives and competencies of the session participants and available computer resources. The reverse is true as well. The objectives and competencies of prospective participants for a prospective training package can be determined to find participants to fill out the roster.

Training package data is accessed using the web services furnished by servlets in the web server. Access is by SOAP/HTTPS protocols and is controlled based on the identity of the accessor (cf. Section Security.)

6.8.3 Resource Manager

Training sessions may require various resources during their operation. Examples include a federation server for HLA simulation-based training or a game server for an MMOG. The requirement for such resources is specified by the training package, but the actual resource to be used needs to be identified and configured into the training session before it can be started.

The resource manager keeps track of the actual and latent resources that are available to satisfy the requirements specified by the training package. Actual resources are resources that are ready for use. Latent resources are resources that can be made ready, but require some action such as starting a server application.

The resource manager is a small, distributed application with outposts on the server machines that can provide the required resources. The outposts can be used to fire up particular applications when necessary. The outposts also compute performance metrics such as ping times to aid in choosing between alternative resource instances.

6.8.4 Session Log

The session log is a list of events that occurred during a training session. It has two components/protocols: The first allows access to the metadata of the logs or record-keeping functions. This is patterned after many of the other components/protocols described in this section and uses SOAP/HTTPS. It allows searching for session logs related to particular members, training packages, time periods, etc.

The second component provides access to the records of a session log. The records correspond to events that occurred during a training session. Contributions to the session may come from any or all applications running in the session. Every record has a timestamp and source to identify it. An additional sequence field allows simultaneous events from a single source to be distinguished.

The data records are stored in the central database as entities. A training application (e.g. a training system) would use this component to add significant events to the session log as they occur. In the same way, observers might add comments to the session log. Later, during the after action review the search and read methods would be used to locate the comments and display the events leading up to the comment.

This interface needs to accommodate a large volume of data and may require significant buffering of that data. For this reason, a protocol with a data representation that is more compact than SOAP will be used. Even so, it is possible for a training system or other application to overuse the session log. Events to be logged should be chosen wisely.

The session log is *not* the after action review (AAR) and does not record the events of an AAR. However, an AAR is likely to refer to events in the session log.

6.8.5 Session Results

Training sessions could have a number of products. Some of the products are incorporated directly into the students' training profiles. Others serve to document the session itself. The primary training session result is the after action review (AAR). While the AAR may be a process, the discussions, comments, observations, arguments should be documented. We have debated whether the AAR process should be recorded or not. If the process occurs over a well-defined interval, it can be easily recorded (as an extension of the session log), but if the process extends over multiple logins, identifying the training session may become problematic.

Details of session results are unspecified. Different types of training may require different types of AAR work products. Those products will be a collaborative effort and may use the DARWorld collaboration tools as well as conventional documentation applications (e.g. PowerPoint).

7 Use Cases

This section illustrates the interactions for various components and API implementations (elements of the previous chapters) in typical cases. Although there are too many use-cases to treat exhaustively, all major capabilities of DARWorld will be showcased in the following examples. Obviously, the use cases in this section are composed of more primitive use cases, and they can themselves be combined to higher-level (super) use cases.

Since pedagogical uses are central to the design and implementation of DARWorld, the following sections mainly focus on the perspective of a typical trainee, the main targeted user of DARWorld.

7.1 Training as an Individual

7.1.1 Vignette

A newly minted platoon leader has just returned from leadership training courses at Ft. Benning. Shortly after returning, he receives a message from his commander with a new DARWorld training assignment.

The announcement email includes a convenient hyperlink to the soldier's personalized DARWorld Web page, which contains his personal training calendar, current news, and much more[5]*. It focuses on making sure he has the information that he needs and knows when to be available for upcoming events.*

Using the calendar, the platoon leader discovers that he has been assigned a series of ten training packages in the "Full Spectrum Leader" software. He knows that these scenarios in these packages will be designed to trip him up, to test the limits of his newly acquired knowledge.

It's not a surprise that, - having used a computer before -, our recruit immediately recognizes the value of DARWorld Object hyperlinks at the bottom of the calendar. These connect him with resource pages built up over time by previous users of the same training packages. There's a series of "walkthrough" descriptions explaining strategies for the first scenario, and a rather lively discussion about which of three popular approaches will be most effective and lead to the fewest casualties. The recruit explores a few of these links and discovers, as is often the case in combat, there is no one simple answer, and our recruit will have learned at least part of this valuable lesson even before attempting to work through the scenario.

Having checked his personal profile for correctness, -it's the first time he's logging on to DARWorld after his promotion and he wonders if his new student records show up- , he clicks on the first of his new assignments and quickly delves into his training ...

Halfway through the first scenario, it becomes clear that the trainee is in trouble. He has mispositioned his forces and is rapidly on his way to losing control of the situation. Recognizing the trouble he's in, the training software asks if he'd like assistance from a subject matter expert. He agrees, and shortly a "reachback" specialist located across the country and is sent an instant message. The specialist clicks on a hyperlink in the message and is added to the recruit's training session. The specialist pauses the software, reviews the recruit's plan, and points out the reasons that he's encountering problems. Unfortunately, it's too late to salvage the operation, because the friendly forces are already largely combat ineffective so the training session is terminated. But the recruit has gained some valuable insight, and has gained confidence as well.

[5] This page performs many of the same functions as Yahoo Groups.

It is late and the recruit is torn between continuing and getting some rest. He decides things will look better in the morning. But, sleep comes fitfully and the recruit is back before dawn and eagerly brings up the AAR. Even if his mission had been completed according to plan, there would still be lessons to be learned and improvements to be made. He knows that the careful review of every area in his professional training will make him a more effective war fighter. It is for this reason that such reviews are so ingrained into our modern military culture.

He decides to look at what went wrong the other night and how others have addressed this challenge. He goes to the AAR link for this scenario and reviews the 3D "interactive recording" of his own performance, looking carefully at a few "3D Post-It Notes" left by the Intelligent Tutoring System (ITS) to point out problems that it had noticed. He then spends half an hour reviewing the AAR for a previous mission, annotated by the player to highlight the key decision points: where the enemy was massing his forces, where he had a weakness in the coverage from his fields of fire, and so on. Armed with this knowledge, the recruit is now far more effective in formulating and executing plans, and, full of confidence, starts the first scenario again...

7.1.2 Sequence of Actions

Login	• The recruit logs into the "DARWorld Command" through the registration service. • The recruit's home page is displayed.
Check Personal Home Page	• The recruit checks his DARWorld. It includes links to the training packages for the ten training missions. • The recruit clicks on links associated with a training package to visit web pages containing "walkthrough" descriptions of strategies, approaches, etc.
View/Edit Member Profile	• The recruit loads his Member Profile using the editProfileServlet. • The recruit checks the settings and closes the servlet.
Launch Training Session	• The user clicks on the training package link for the training package that caught his eye earlier. • The user clicks the "start training" link to start. (Content-type of application/DARWorld-launcher) • An ad hoc training event is created and the launcher is started. • The launcher interprets the contents, which contains all the information need to run the training session. In particular, it names the training package and all the bindings to define the unbound parameters of the training package. • The launcher fetches the data of the training package. • The launcher applies the bindings to the training package and launches the training system with the fully bound training package. • The launcher establishes connections to the other resources needed for the training session. • The launcher tells all resources to start.
Reachback	• The training system asks the recruit if he needs assistance, the recruit confirms. • The training system uses IM to contact the SME, copying a link to the current training event in a message. • The SME follows the link and visits the training event home page to. • The SME might browse the training package, the training profile of the student, etc. • The SME adds himself to the training session and his training system instance is launched. • The SME pauses the training session, scans the session log, views the

	situation from a God's eye view. • The SME responds via IM providing feedback and suggestions.
Terminate Session	• The recruit terminates the training session.
AAR (own session) [training system-specific]	• The recruit visits the training session home page and clicks on the AAR link. • The recruit chooses the "playback" option and watches the recording of his session, placing the camera in specific locations. • The recruit chooses the "SME feedback" option. • The system presents the IM log to the recruit. • The recruit chooses the "ITS feedback" option. • The system presents the marked parts of the recording including the annotations by the ITS. • The recruit browses through this recording.
AAR (other session)	• The recruit searches for other sessions using the same training package. • The recruit visits the training session home page of one of the search results and clicks on the AAR link. • The system checks his permissions and grants access to that specific AAR.

7.1.3 Use Case Variations

This use case described a scheduled training session that had been arranged by the recruit's commander. Variations are that our up-and-coming leader logs on to DARWorld and browses through the system querying the database for training packages that are relevant to his training objectives, and selects a package that appeals to him. He then can also decide if he wants to practice under his real name or if he wants to use an "alias", so that the score does not affect his official report.

In another form of possible reachback/collaboration the recruit calls up several friends using Instant Messaging and tells them what happened. Together, they draw out a map on the DARWorld Whiteboard and he sees what he's been missing.

The reachback can vary depending on the role the reachback specialist takes on. Based on his judgment, the desires of the trainee, and the nature of the training session, he can be a passive observer, simply watching the action and moving silently through the world, not affecting anything, maybe only annotating events. Or he can play the part of an instructor, manipulating the situation in order to illustrate a particular point. Or he can become an active participant, taking over control of OPFOR or BLUEFOR and demonstrating by example how best to deal with a particular situation.

7.2 Team Training

7.2.1 Vignette

Our intrepid platoon leader is not alone in DARWorld, as we have already seen. He has access to friends and real-world teammates, and also to specialized training resources such as reachback personnel. The following use case explores how the platoon leader might interact with other DARWorld participants.

In addition to taking personalized training as an individual as described above, the platoon leader has also been participating in some larger-scale multi-user training exercises. One of these is a battalion-level trainer that links several smaller simulations together for a force-on-force combat experience. More than a dozen real-life platoon leaders, plus half a dozen captains and a variety of automated entities, join together in a monthly challenge match that's designed to keep their cognitive skill sets fresh and give them an opportunity to explore new tactics.

Our platoon leader receives an email announcing an upcoming event for his battalion combat team. He signs on to the DARWorld Web site, reads the briefing, and prepares to enter the scenario. Some specific activities that this particular user might want to engage in while on the site:

- *Review the "Profile Page" for the team to see how its performance compares with that of other teams that are similar (perhaps listed in order of geographical or organizational "nearness")*
- *Update the website with personal descriptions ("war stories") of a previous game.*
- *Arrange a live, online briefing walking each of the other team members through a few important tactics he's identified (he can't do this in person, because two of his teammates are in Pakistan).*
- *Leave a few tips and tricks on the team's private Wiki page summarizing the briefing, because two of the team members had drawn KP duty and weren't available at the scheduled time. They can then review the briefing at a later date.*

On the day of the event, in addition to the players, several dozen observers are present (electronically) as well, recording their observations of the teams' performance and preparing their own after-action reviews. These people are not seen or heard by the participants in the scenario; they are effectively invisible.

Unfortunately, two teammates of our platoon leader have been reassigned and deployed the day before the exercise and are unavailable, and there is no way to find substitutes. However, DARWorld is able to fill the empty spots with AI players.

The exercise is started and terminated by the Training Supervisor.

After the event, everyone gathers for a good-natured yet hard-edged evaluation of the situation. That online briefing (text messaging plus visuals) then goes into the online archive for later review. Depending on the scenario, the AAR may include various annotation tracks prepared by observers, instructors, and/or participants. If it is important for these to be seen during the AAR, the AAR can be scheduled such that the other participants have an appropriate amount of time to prepare their annotations. These can then be viewed by all participants, locally, as they are listening to the AAR.

Following the event, other interested personnel (the commanding general's staff, for example), can review the AAR and the annotations, which by then will have been fully indexed by the DARWorld Search Engine.

7.2.2 Sequence of Actions

Login	• The platoon leader logs into the "DARWorld Command" through the registration service.
Personal Home Page Pre-Training Session Activities	• The platoon leader checks personal home page to read the briefing for the event. It includes a link to the training package for the upcoming mission. • He clicks on the link associated with a training package to visit web pages containing "walkthrough" descriptions of strategies, approaches, etc. • The platoon leader clicks on a "Team Profile" link to bring up a page that provides information about the team and links to other team profile information allowing for team comparisons. He can then assess/review the strengths and weaknesses of the team and compare this team to other teams. • On the training package page, there is a place where new "war stories" and strategies can be added. After that information is added, it is saved in the backend database associated with this training package. • The platoon leader returns to the team's private home page. • From that page, he can use the Wiki to arrange and execute a live online briefing for other team members. • The platoon leader adds tips and tricks to the Wiki as a summary and for review by team members that didn't attend a briefing.
Launch Training Session	• The trainees bring up the training event page and indicate their readiness to participate (they are now "in the lobby"). • Observers also bring up the training event page and add themselves to the event as observers. • The training supervisor brings up the training event page and verifies that everyone is ready to participate. Realizing that two of the participants have been reassigned are unavailable, he assigns AI players to their roles. • The Training Supervisor clicks the "start training" link to start. The content-type of application/DARWorld-launcher implies that the content is to be interpreted by the DARWorld launcher. • The launcher interprets the contents, which contains all the information need to run the training session. In particular, it names the training package and all the bindings to define the unbound parameters of the training package such as the IP addresses of the other participants. • The launcher fetches the training package. • The launcher applies the bindings to the training package and starts the training session by: o The launcher establishes connections to the other resources needed for the training session (participants platforms, AI platforms, and training system servers) o Once connections are established and all resources and training systems indicate they are ready, the lead training system is chosen. (The choice is simple and based on IP address.) o The lead training system tells all resources and training systems to start.
AAR (own session) [training system-specific]	• The platoon leader visits the training session home page and clicks on the AAR link. • He chooses the "playback" option and watches the recording of the training session, placing the camera in specific locations. • Using a web-based interface, the platoon leader adds annotations associated with particular events during the training session. This information is passed to the session log on the DARWorld Backend. • At the designated time, all participants and the instructor can review the events of the training session (and annotations).

AAR (other sessions)	• Other interested personnel (e.g. commanding general's staff) are passed the AAR link in an email. They click on that link. • The system checks permissions and grants access to that specific AAR. • They review the session and search for areas of interest using the "Session Results" interfaces.

7.2.3 Use Case Variations

In preparation for this large-scale exercise, each team wants to practice. The DARWorld system lets soldiers form their own groups to practice. These can be based on their actual unit assignments or on mentoring relationships they've built up over the years (perhaps a Lt. Colonel will join them and teach them some lessons from his latest deployment, or a few captains actually in the midst of a combat assignment would help them prepare for their own upcoming overseas duty). The DARWorld Calendar helps schedule such events.

During a practice exercise, there are some empty spots to be filled. The DARWorld lobby allows players to join a team exercise.

A very informal way of joining a team is the "ask your buddy to join" method that allows a DARWorld trainee to use IM to contact his buddies that can then follow the link provided via IM and sign up for the roles they wish to play in the session, negotiating with IM as needed. When all roles requiring human participants are filled, it becomes possible to start the session.

7.3 Session Setup

7.3.1 Vignette

A company commander has selected a number of personnel for promotion, and has replaced several more who rotated out of active duty. Now these personnel must be trained for their new positions.

First, the commander reviews the training records of the personnel and orders them to attend the standard training courses now in use (classical "high fidelity, low frequency" training). One new component of the training is an ongoing education program; once the trainees return from their classroom exercises, they receive continuing online education and readiness training within DARWorld.

The commander logs in to DARWorld through the Web browser on his desktop or portable computer, and then designates specific training objectives for certain personnel. Although they have succeeded in their schoolhouse assignments, it is clear that they are still inexperienced in the application and pragmatics of their new responsibilities: They need practice applying the skills that they have learned! The commander therefore makes a formal request, using the scheduling feature of the DARWorld Calendar, for them to engage in specific ongoing training. Furthermore, in order to strengthen the working relationships of his existing personnel, he also assigns several of them to participate as observers and coaches in certain of the training exercises. This, too, is handled through the DARWorld Calendar.

- *Each person on the team receives a DARWorld message announcing the assignment. These users can sign on to their personal DARWorld "home page" at any time to check their schedules, review package materials, and more (detailed examples of these activities are spread throughout the various use cases).*

- *In addition, the company's official Training Supervisor receives a message as well. This person will be responsible for scheduling events, coordinating joint exercises with various personnel, and ultimately for ensuring that the trainees complete their DARWorld assignments in a timely manner, in accordance with the commander's direction.*

7.3.2 Sequence of Actions

Login	• The company commander logs into the "DARWorld Command" through the registration service.
Assigning Training Sessions	• The company commander requests to bring up Member Profile information for desired subordinates. • The system checks his permissions and grants access to the requested member profile information. • From the member profile page, the company commander edits the training goals and saves the profile. • The member profile information is saved in the DARWorld Backend database. • The company commander checks the subordinate's calendar and schedules a new training event aligned with the updated training goals. • He sends an email to the subordinate with the order for the new training event. • He also sends email to required observers providing the training session link, trainee information and observer goals.

7.3.3 Use Case Variations

Setting up a training session, (creating a training event) means parameterizing a training package. The training package specifies what scenario is to be used, what roles need to be filled what training objectives each role addresses and what training achievements (competencies) are required. It also specifies what training system (training software) is to be used and what other resources are required. Parameterization is the process of assigning values to (binding) some of the unspecified variables of the training package such are who is to fill particular roles. Some values such as IP addresses for the players are filled in at the last moment in the "lobby" as players indicate their readiness to participate.

The use cases vary depending on which steps of this parameterization are done by a human (e.g., the training supervisor, or in the vignette, the commander) and which steps are done automatically by the DARWorld system using knowledge about the current situation and request made by the user.

7.4 Authoring

7.4.1 Vignette

The captain sits down at his desk. He has just returned from a short but difficult deployment overseas. He has learned a great deal in his travels. Much of this will be invaluable to his compatriots who are on their way out to the field in a few months. How can he best share this information? He remembers his training in DARWorld and now he sees a number of areas where he can create custom training packages.

He decides to focus on a new technology that was deployed on an experimental basis with his unit: the Micro Air Vehicle, or MAV. This tiny, Frisbee-sized device came with a handheld touch-screen controller, but only minimal instructions on how to employ it in practice. He will prepare a series of lessons to help new users make the most of this new tool, based on the lessons that he recently learned. Ultimately, he hopes that this curriculum will lead to the creation of a new, formal set of TTPs (Tactics, Techniques and Procedures) for future deployments.

Inside DARWorld, anyone can create training packages and scenarios with a modest amount of effort. This is one of the key features of the system. Training packages and scenarios will often be built by instructors or by members of a class studying a particular subject. But individuals in other capacities can build them for their own purposes. Our commander knows that he can easily put together a quick scenario and build a training package using it. Or, if he doesn't have the time, he could just as easily assign a junior staff member to take care of it. The learning curve should be low enough that all interested parties can make scenarios if they choose to do so.

There are official training packages, and there are individually developed packages that are also available within DARWorld. Official training packages are approved parts of the training regimen. They may be used only in classroom situations, or they may be posted in common areas for all to use. Sharing will be encouraged. In many cases, an effective training package for a particular purpose can be created very quickly if the scenario creator begins with one that already represents the desired terrain or type of operation. Oftentimes, the bulk of the time will be spent in writing the OPORD and not in the mechanics of scenario creation. The pedagogical leverage from such situations is tremendous: for little more effort than what would be required to create a traditional paper-and-pencil scenario, the scenario creator can build a dynamic, explorable cognitive training simulation.

- *Our commander signs on to DARWorld and searches for available company-level training packages using a particular tool that he has used many times in the past. He quickly locates an updated that he used during his training a year earlier. Later instructors have added a number of useful improvements to the OPORD and tutorial elements.*

- *From the home page of the training package. He clicks the "edit" link in his browser to invoke the training package editor. In the training package editor, he selects the "Edit Scenario" menu item. DARWorld checks his version of the training system, which turns out to be six months old, and updates it to the current code release. It then copies the scenario to his local machine and launches the training system scenario editor.*

- *The commander makes a few quick changes to the order of battle for the friendly forces, adding in several UAV units already supported by the code. Although these aren't quite the same as the MAVs in appearance, they'll work just fine for training purposes, at least for now. He goes back to his Web browser, clicks on the Author link for the training system, and sends the company a quick note requesting support for MAV-type units in the next release.*

- *He returns to the training package and, since he is not allowed to update the original training package, he saves the original training package with the updated scenario in his private training package area. After saving off his work so far, he loads up the OPORD and begins to customize it for his needs. He quickly realizes that he needs to simplify the mission a bit, and expand the role of the OPFOR slightly in order to better match conditions where he was deployed. He emails the OPORD to a few of his friends using the DARWorld "Buddy List" (one of whom is still overseas), and integrates their suggestions during the following day.*

7.4.2 Sequence of Actions

Login	• The company commander logs into the "DARWorld Command" through the registration service.
Launch Training Package Editor [Training system Specific]	• Using the "DARWorld Command" Matching Service, the company commander locates the training package with the scenario to modify. • He clicks on the "edit training package" link to launch the training package editor and views information about the scenario. • He selects the "Edit Scenario" menu item and the launcher launches the scenario editor of the training system. • The launcher applies the bindings and launches the training system

	scenario editor. • The launcher establishes connections to the other resources needed for the editing session. • The training system tells all resources to start.
Modify Scenario [Training system Specific]	• Using the training system scenario editor, the company commander modifies the scenario. • Back to the scenario information page, he clicks on the author link to send the scenario author an email indicating the changes being made.
Completing Scenario Modifications	• Once the changes are done, the company commander saves the training package with the modified scenario into his private area of the DARWorld Backend database. At this time, the flag for official DARWorld training scenario is not set. • He sends an email to group of reviewers for feedback.

7.4.3 Use Case Variations

There are various kinds of authoring relevant in DARWorld: scenario authoring; content authoring; event authoring; building models automatically from raw data to be incorporated into scenarios. Many of these depend on specific tools to be built in the future.

DARWorld supports these authoring variations by providing: a standardized feedback channel to the authors; means to search, annotate, and publish scenarios (e.g., Google-style DARWorld Search).

Scenario creation doesn't end, however, with this initial release. Along with the posting of the scenario, special DARWorld Wiki pages will be created to provide a place where students can post feedback or suggestions about the scenario. Message links may also be provided, allowing them to contact the scenario creator via email or instant messaging, to discuss issues they may have. In addition, users will be able to browse through archived AARs using the scenario, benefiting from what others have learned from it. Ultimately, this will probably result in the creation of new and improved versions of the scenario or its related tutorial content. Our commander's efforts are only a starting point. The initial scenarios that are created will most likely spawn a significant number of variations and explorations, the best of them eventually working their way into the regular training regimen for MAV operations.

8 Data Management Design

8.1 Data Types and Distribution

8.1.1 Member and Group Data

Member data consists of all the information held by DARWorld about a member and ranges from contact information such as name, address, telephone, and email to training objectives and achievements and raw test scores. This data is held in a relational database, but accessed with the container-managed persistence feature of the J2EE container that performs access control and provides database independence.

Some member data is manually maintained through web forms posted by servlets of the web server. There is nothing esoteric here – just standard web design. Other member data is updated as members train. Records are kept of the training packages a member has completed, the outcome of the training, etc. Since training sessions may involve multiple members, the training session data is recorded separately and summarized and referenced by the individual member data.

8.1.2 Training Content

Training content is the generic term for all the data used for training. It ranges from simple web pages to complexes of training packages, scenario files, prerequisites, etc. DARWorld distinguishes content from metadata that weaves the content into a logical pattern to address the training objectives of its members.

The most complex data structures center around the training package. The complexity arises because of the multiple criteria it must satisfy. A training session requires a number of students, instructors, subject matter experts, and other members serving particular roles. It also requires certain computing services for game engines and federation servers. It may also allow a variable number or optional participants such as observers. Furthermore non-human players that may also require computing services may fill some roles in the training. A training package must describe the requirements for these roles in a way that makes it clear that a particular training package is good for training a particular set of students if the other requirements can be satisfied.

From the learning management system viewpoint, selecting a training package is a complex form of sequencing. The individuals have certain training objectives and have achieved a certain level of competence or experience and the sequencer must decide what they should do next.

Further complication arises because the training package must also address a similar set of issues for team training.

8.1.3 Session Data

The session data are all the artifacts that are created during a training session. These range from simple record keeping of who participated, when did the training occur, what training package, where the participants were to detailed session logs of what happened, what comments were recorded for after action review, replay data, etc. Included is an assessment of the performance of the participants and the raw data to support that assessment, if any.

8.1.3.1 Session Record Keeping

Session record keeping is straightforward. For each session a record is made of what training package was used, when the session began and when it concluded. The reason for the conclusion and the status of the session at its conclusion is recorded. The member id of who filled each role specified by the training package is recorded with his location (IP address) and the times of his participation. The latter may be different from the session times in the case of intermittent participants such as reach-back experts. If a non-human player filled the role, the version of the software playing the role is recorded.

8.1.3.2 Session Log Data

The session log receives events from multiple sources: instructors, intelligent tutors, students, reality substrate, etc. Each event recorded in the log is uniquely identified with a source and timestamp plus a sequence number to define a sequence for what would otherwise be simultaneous events from a single source. Beyond that, an event may have arbitrary attributes such as position (in some coordinate system). A standard set of attribute names will be defined for certain generic attributes, but the set is open-ended. Any attributes not in the standard set are private to the source of the event and that source is responsible for any interpretation of the value of that attribute.

8.1.3.3 Replay Data

Replay is highly idiosyncratic. Ideally, replay data would permit the reconstruction of an arbitrary portion (in time and space) of a training session from one or more arbitrary points of view. Replay data is usually thought of as a file. If there are multiple processes involved, there may be multiple files. These files can be very large and are frequently not useful beyond the immediate situation, e.g. during an AAR immediately following. Therefore, moving the data to a central repository is not automatic, but happens when certain "save for session review" style actions are taken.

8.2 Persistence

The DARWorld servers are responsible for long-term storage of the data in its database and files. Any data item that refers to another must do so explicitly. Implicit references to other data are considered programming errors and there will be no guarantees that the referenced data will be available forever. The references are entirely held in the database. Data files must be explicitly referenced from the database. Files not so referenced may be removed.

An example might make this clear. Consider a session log. The record-keeping data refers to a training package and members. These are explicit references to the other entities and so the other entities will persist for as long as the session log persists.

Cleanup is a standard garbage collection problem. By requiring all references to be explicit, all non-garbage is readily identified. Only the garbage is removed; non-garbage remains.

8.3 Disconnected Operation

The same database design principles that allow for efficient data persistence also enable disconnected operation. By "disconnected operation" we mean running DARWorld training when network connections to the main DARWorld servers are not operational. The design allows the data that supports other data to be identified so such data can be retained during a garbage collection. The same "other data" constitutes what is implicitly required to use the data offline.

The preparation for disconnected operation requires selecting the individuals who might participate in training while disconnected. Then the training packages to be used are selected. The selection could be explicit or the selection could be implicit and consist of the training packages that address the training objectives of the group of individuals that will be training in the disconnected DARWorld. Since DARWorld already has the capability to select relevant training packages for individuals based on their training objectives and competencies, the implicit selection of training packages is almost trivial.

The member records for the selected individuals (and groups, if any) plus the selected training packages and all the "other data" required to support them is either bundled up into a "take away" installer for later transfer onto the disconnected machine(s) or directly installed onto the machine(s) that will become disconnected.

When disconnected operation is complete, the main database should be updated with the training results and other changes made while disconnected. There are substantial security issues with doing this since important data that would normally be physically secure may have been installed on personal laptops or other physically exposed machines while disconnected. At one extreme, no data would be updated; the only "update" would be to the users' minds. At the other extreme, the changes would be unconditionally accepted. In between, the updates would be made according to access privileges of the individuals entrusted with the sensitive data. Instructors would have responsibility for safeguarding the training records and would be allowed to modify the permanent records using the disconnected training results. Students would be allowed to update their personal profiles and portfolios.

8.4 DARWorld Object Reference

Many elements of DARWorld will be accessible with a URL. For example, an AAR link could be emailed to someone else. Clicking on the link would open up an AAR browser that would allow the AAR to be scanned. Interesting events could be opened starting a training system initialized to the scenario and replay data for the session. The context of the event could be examined from various viewpoints.

Every type of object that can be referenced this way will present a unique challenge. Consider, for example, an experience that a user has that he wishes to share with a colleague. What is the experience? Perhaps it is the user's state of mind – very difficult to encode into a URL. Maybe it was the training package that was used – difficult to use since the recipient will need to recruit a similar set of participants to share the experience, but a perfectly fine interpretation of "experience". Perhaps it was the scene of a triumphant success in a test of strategy – a reasonable expectation. Each of these, while they may share certain parts, will require designing a data representation that captures both the common aspects of the various experiences and allows the expression of the unique aspects of each experience.

The essential architectural requirement is to recognize the need to factor the data into a succession of parts that can be assembled to represent the experiences we choose. The training package is a sub-assembly of some of these parts that then becomes a part of more specific experiences such as a scene in a simulation. A scene also has additional parts such as replay data that represents the simulation evolved to produce the scene. A scene could be static or dynamic. A static scene is a snapshot. It can be viewed from various points-of-view, but cannot move in time. A dynamic scene has additional information allowing it to be viewed in time.

More mundane objects also exist. The goal is to be able to compose reports, summaries, BLOGS, IM messages, etc. that include these references so that navigation is simple.

8.5 Security

No Internet system can survive for long without careful consideration of its security requirements. DARWorld is no exception. A basic premise is to protect everything and assume nothing. All communication will be encrypted and a well-defined access control scheme will permit services to limit access to authorized members. The access control scheme is intended to limit access; it is not intended to detect program bugs. Applications are still responsible for changing the data correctly.

Access control is based on who the actor is, what action he is trying to do, and what data he is trying to act upon. Authentication is the process of determining who an actor is. Authorization determines the ability to perform actions on data.

8.5.1 Access Control

Access control can be divided into authentication and authorization. Authorization is an enumeration of allowed or disallowed actions. Each of these is described below.

8.5.1.1 Authentication

Authentication is the process of determining the identity of a principal (an individual or entity). In a security system using a public key infrastructure[PKI] authentication is based on certificates. A certificate is a (digital) document signed by a certificate authority (CA) containing the identity of the principal and a public key. The principal has the private key corresponding to the public key and engages in a protocol in which the certificate is sent and information is encrypted in the private key to allow the remote party to securely establish the identity of the principal. The protocol may also authenticate the principal at the other end of the connection so both parties know with whom they are communicating. SSL and TLS can perform authentication when a connection is established and that is the method that will be used.

8.5.1.2 Authorization

Authorization describes the actions that a principal is allowed (has the authority) to do and consists of two parts: what the action is and what the object of the action is. There are two conventional approaches to establishing the authority of a principal. A certificate may include "extensions" that serve to enumerate the capabilities of the principal and can therefore define the authority of that principal.

Alternatively, authority can be obtained from a separate database listing authority versus identity. The tradeoff is flexibility in altering a principal's authority versus simplicity in determining authority. When authority is based on certificate extensions, new certificates must be issued when a principal's authority changes. Having a separate authority database requires, in principle, a reference to that database prior to authorizing any action. This burden can be mediated to an extent by caching the authorization information, but this makes changing the authority of a principal problematic.

DARWorld will use an Authorization Service to determine the authority of a principal and certificates will be issued for relatively long durations. A system of leases and timeouts will achieve a reasonable compromise between excessive use of the authorization service and excessive delays in altering the authority of a principal. In principle, changing the authorization of a principal requires a delay to allow leases to expire. This delay is tolerable because changing the authorization of a principal is infrequent.

8.5.1.3 Authorized Actions

The representation of authorized actions can range from a single bit (all actions on all objects (or not)) to arbitrarily fine-grained control of every possible action on every object. DARWorld leans toward the latter, allowing arbitrarily fine-grained control with the individual services determining what the appropriate granularity is for its actions and objects.

The authorization database is logically a collection of sentences of the form:

<Role> (can/cannot) <verb> <object>

Every principal has one or more roles. Every principal has a role equal to its member id. Principals also have roles equal to the group ids of the groups they belong to. This allows arbitrarily fine granularity of authority. But in most cases, authority will be defined in terms of a standard set of roles such as "student". The verb is arbitrary, but it is necessary to avoid naming conflicts between independently developed software. To avoid making the authorization database an administrative nightmare, verbs will be administratively "managed" and relatively few in number. There is a wild card verb meaning, "do anything to" or "has all access to". Objects can be specific objects, classes of objects, or all objects. The wild card provision is especially useful for specifying default access for a container (see below). Objects have access control lists (ACL) that list the allowed roles for various verbs.

We speak of controlling access to "objects" while the DARWorld database is a relational database. Mapping objects onto a relational database is reasonably straightforward as long as the data structures are not too complex and there are relatively few "classes" of objects. We won't dwell on this subject any further here. It is mentioned only to set the stage for describing access control lists.

Access control lists are kept in the database. The database is factored in ways to facilitate access control. For example, instead of storing member information in a monolithic single table, several tables are used where each table corresponds to a sub-object of a member such as his contact information, training objectives, training achievement, and so on.

Objects form a containment hierarchy where some objects contain others. The DARWorld object containment hierarchy, for access control purposes, is a simple tree. Each container may establish default access control for its contents. Access denial is cumulative from the top down. That is, access is denied if any of the containers enclosing an object deny access to that object. Conversely, access approval is cumulative from the bottom up. Another way to say this is that a more specific approval overrides a more generic denial. The default access control of the root container is to deny all access to all objects.

8.5.2 Communication Security

A secure socket layer (SSL) provides transport level security (TLS) as well as authentication through the use of both client and server keys and certificates for all connections (with minor exceptions). This serves two purposes: it protects the privacy and integrity of the data and establishes the identity of the principals involved providing accountability. The major impact of this is that all users will need to obtain one or more certificates to use DARWorld. These certificates will be issued as part of the membership application process and may be held in certificate storage on the user's machine or stored in hardware tokens to be plugged into any machine he uses.

Although web services are evolving toward the use of message level security, the transport independence it supplies does not justify the additional complexity. So, there is no plan to provide message level security for DARWorld.

8.5.3 Public Key Infrastructure

The public key infrastructure (PKI) will be assembled from COTS, GOTS, OOTS software tailored as needed. Its major components are a certificate authority (CA) and certificate revocation list server (CRL server) and a number of registration authority (RA) interfaces. In addition, infrastructure java libraries will help configure applications to access the user's certificate store.

We will use EJBCA [EJBCA] for a certificate authority. EJBCA is an open source, fully functional Certificate Authority (CA), written entirely in Java and based on J2EE technology. It includes everything listed above though it must be configured for DARWorld. EJBCA requires an LDAP server and openldap will be used for that purpose.

9 DARWorld Administration

DARWorld will ultimately be an extensive distributed system that must be maintained, as any such system must be. There is not much novelty in system administration. For the most part, conventional approaches will be used. An exception is automated client software maintenance and that is discussed below. For completeness, we list the specific areas of system administration that ultimately need to be addressed.

We divide DARWorld administration into a number of areas based on aspects of the system: member, training content, and server and client administration. In addition, there are a few over-arching administrative facilities that apply to most of these system aspects.

9.1 Member Administration

Member administration is nothing more than additional variations on the theme already established for other member and group functions. Administrators are simply DARWorld members with roles giving them access that allows them to perform their job. There is a bootstrapping issue to solve the chicken and egg problem. Simple database scripts will create administrators that can create/approve other administrators. The architecture does not dictate policy. Over time, policies will emerge or be imposed that administrators follow and additional servlets or other facilities may be developed to automate the procedures that administrators follow and eliminate some of the detail required to adhere to policies.

Administrative groups will allow the member space to be segmented into manageable pieces. Ordinary web pages will guide new members to the appropriate administrative group. There is no design for this aspect of DARWorld, yet.

For security purposes, administrative activity will be logged in detail by the servlets performing the actions.

Defects in or enhancements to the member administration process will be entered into the bug reporting system. Informal channels may be used in some situations such as a member inquiring about the status of his request, but real problems and enhancements should be formally documented.

9.2 Training Content Administration

Training content administration happens at several levels from scenario development, creating training packages, defining training objectives, creating lessons, defining prerequisites, and maintaining the training catalog. Much of this is simple servlets and web pages to query, navigate, and edit the training content. Some tasks require specialized software such as scenario and training session editors.

Training feedback such as problems with existing training packages or the need for new packages and scenarios will be recorded in the DARWorld bug reporting system. Informal feedback may use instant messaging or email, but formal documentation of problems and needs by the sender or receiver of such informal messages is the norm.

9.3 Server Administration

Server administration is conventional with web servers, databases, game engines, and a few specialized daemons. Servers may be collocated or dispersed depending on the performance/redundancy trade-off. It should be feasible to farm out the server administration task if desired.

While server administration may be conventional, that doesn’t mean it is trivial. There may be substantial issues regarding database, website, and package updates. This document offers no guidance to such issues at this time.

9.4 Client Administration

The goal is to automate most of the client side maintenance with automatic software installation and updates. To bootstrap the process, it will be necessary to install applet software to enable access to the client hardware. ActiveX could be used, but that may be undesirable from a security point of view.

Beyond the initial toehold, additional software can be installed as needed including software to perform version checking and updates. These subsequent software installations and upgrades can occur automatically (subject to user approval) or can be explicitly initiated by the user. Details of this process will depend on what software is chosen or created for this purpose.

9.5 Bug Reporting

A bug-reporting system such as Bugzilla will be used to report and track bugs and enhancement requests. Numerous products and components will be named to guide the reports to the relevant groups.

Bugzilla is a placeholder and may not be the ultimate choice because it has usability issues; the user interface, particularly the query interface, is clunky and non-intuitive because it sacrifices ease of use for flexibility.

9.6 Error and Exception Reporting

DARWorld software will make extensive use of conventional error logging systems. For Java applications the standard java.util.logging classes and interfaces will be standard. For non-Java applications, the standard will be an XML-based format consistent with the format of java.util.logging.XMLFormatter output. Software developers are expected to log all significant issues their software encounters and as much additional debugging information as they feel may be useful. The error logging system supports a number of logging levels based on the severity of the problem or the detail needed.

Errors and exceptions will normally be logged to files on the machines where they occur and uploaded as needed for bug reporting purposes. Applications can also be configured to send output above a certain level to a data server to be recorded in the database. Such automatic error reporting will help spot trends and augment standard bug reporting methods.

10 Conclusions

The matrix in Figure 10 pairs the five characterizations of the DARWARS vision presented in Section 1 against some of the architectural features of DARWorld. X's show where features are particularly relevant to reaching the vision.

Architecture Element / Vision Goal	User Management	Learning Management	Social Communication	Training Package	Session Results (e.g. AAR)	Explicit Object Dependencies	DARWorld Object Reference	Integrated Database	Auto-deployed Software	Generic LMTS Architecture
Universal	X		X	X						
Continuous and Persistent	X	X			X	X		X		
On Demand			X			X	X	X	X	
Engaging			X	X			X			
Effective		X		X	X	X	X			X

Figure 10: Reaching the DARWARS Vision

The DARWorld user management system ensures that universal access is possible wherever a user might be, if he has a computer and a network connection, he can perform his role in DARWorld whether that might be as a student or trainee, and instructor, content author, or a simple onlooker. With a little foresight, the disconnected provisions of DARWorld even allow universal access when network connections are substandard. The user management system makes what the user has done persistent and continuously available.

The learning management is closely tied to the user management system, but focuses on the training material that is available. By design, such material is always available. Indexing information maintained by DARWorld ensures that the most effective and relevant training packages can be found and used.

The social communications fabric of DARWorld allows members to interact with others regardless of their location or when they might choose to participate. The social interplay adds interest and increases training effectiveness by the sharing of experiences and lessons learned.

Training packages allow experiences to be tailored to suit a wide range of member skill and experience. Training packages provide a focal point for comparing skill and experience and provide a basis for training ladders or other competitive activities that engage the students beyond what they might otherwise do and increase the system's effectiveness.

Recording session results such as after action reviews and making those results available whenever a DARWorld member needs them is crucial to effective learning and assessment of that learning. There is even an element of "engaging" here, as the results from training sessions become fodder for a member's portfolio for promoting upcoming training events.

The use of explicit object reference has an indirect effect. It is an enabling feature that will never shine on its own, but is crucial in the areas indicated by the matrix.

DARWorld object references are all about ease-of-use. Obscure navigation tricks or being required to provide information that DARWorld should already know does not bog down users. Instead, they can focus on what is interesting and effective when they want.

The integrated database of DARWorld insures that the data for any DARWorld activity is available and consistent across all activities. It also means that the data to support any activity is available.

The auto-deployment of software allows on-demand access. No CDs are required and no manual installation procedures need to be performed.

Finally, the generic training system architecture allows the effectiveness of training to be enhanced by enriching one training experience with additional instruction.

11 Appendix

11.1 Learning Management Specifications, Standards, and Interoperability

There are currently a number of organizations, such as IMS, ADL, IEEE, ISO, ARIADNE, etc. that are working to develop specifications and standards for e-learning content and frameworks for interoperability between learning management systems. We describe each of these major efforts, and briefly describe their objectives and illustrate if and how their specifications fit into the DARWorld architecture.

11.1.1 ADL

The Advanced Distributed Learning Initiative[6] (ADL) is a program sponsored by the Department of Defense and the Office of the Secretary of Defense (OSD) and is a collaborative effort between government, industry, and academia to develop a new approach to creating, delivering, and managing learning on a global scale. It is designed to enable the interoperability among learning tools and course content. This is made possible through the use of common, open-architecture standards and the convergence of computing, communications, and information technologies.

The ADL strategy is to: exploit existing network-based technologies; create platform-neutral, reusable courseware and content to lower costs; promote widespread collaboration to satisfy common needs; enhance performance with emerging and next-generation learning technologies; develop a common framework that drives the commercial off-the-shelf cycle; establish a coordinated implementation process; and develop common standards and guidelines.

11.1.1.1 SCORM

In 1999 the DoD was asked to take the lead, in conjunction with other federal agencies and the private sector, to develop a common specification and standard for technology-based learning. The Sharable Content Object Reference Model[7] (SCORM) was developed as a way to integrate and connect the work of these organizations. ADL developed the SCORM to incorporate many of the emerging standards and/or specifications into one common reference model.

The SCORM is built upon work originating in other organizations such as AICC, IMS, IEEE, ARIADNE and is designed to create one unified "reference model" of interrelated technical specifications and guidelines designed to meet DoD's high-level requirements for Web-based learning content. The SCORM includes aspects that affect learning management systems and content authoring tool vendors, instructional designers and content developers, training providers and others.

SCORM includes a Content Aggregation Model (CAM) which specifies how learning content should be constructed in order to be reused, and a Run Time Environment (RTE), which specifies how content is to be launched and leaner progress tracked and reported back. The SCORM assumes a suite of services often called a Learning Management System (n LMS), a Learning Content Management System (LCMS) or Computer Managed Instruction (CMI) System.

---

[6] http://www.adlnet.org/

[7] http://www.adlnet.org/index.cfm?fuseaction=scormabt

The CAM uses the IEEE LTSC LOM specification for describing metadata for the learning content. The metadata describes what the content is, who owns it, what costs (if any), technical requirements, educational purpose, etc. Additionally, the CAM specifies the XML bindings for the metadata, defining how to represent the tags in XML. The CAM also defines how learning objects and their metadata are to be packaged together. This Content Packaging Specification originated from IMS. SCORM version 1.3 introduced a standardized method for representing the sequencing and navigation of learning activities.

The SCORM was designed for creating, delivering, and managing web-based learning content, and is well suited for that purpose. DARWorld supports the learner's experience in ways similar to SCORM, and necessarily has elements of a Learning Management System. However, DARWorld is designed to support persistent large-scale, often simulation based, team activities and therefore must provide functionality not yet found in existing SCORM implementations; e.g., team formation, teamwork assessment, real-time coaching and feedback, simulation as part of the learning context, use of avatars and active scenario manipulation to create events, multiple assessment and tutoring modules, shared AAR, persistent annotated experiences. None of these issues are a surprise to those members of the SCORM community concerned with future extensions to SCORM. Research is beginning to explore solutions to some of these limitations (e.g., work is currently underway to find methods for incorporating simulation as a learning content object). We expect to remain in touch with current work exploring Web service architectures for SCORM, which have the potential to enrich the types of control and evaluation possible within the SCORM framework. We see DARWorld as providing important test cases and potential solutions that will aid the evolution of SCORM.

We acknowledge the value of being able to integrate existing SCORM compliant material in DARWorld by including the ability to embed material as one of the choices available to a learner, alongside the immersive simulation based experiences that characterize DARWorld primary content. We also provide architectural support for training system developers to incorporate SCOs as part of their instruction – thus reaping the benefits of reusable content.

11.1.2 ARIADNE

ARIADNE[Ariadne] is a European Union project focused on building tools and methodologies for producing, managing, and reusing computer-based pedagogical elements and telematics supported training curricula. To support this effort, they are involved in specification efforts involving educational metadata. Since 1997, ARIADNE has been working as part of the IEEE LTSC committee, cooperating with IMS in developing the IEEE/LOM (Learning Objects Metadata) 6.3a working draft. Additionally, ARIADNE is cooperating with the ADL project, whose SCORM specification relies on the LOM metadata.

To the extent that DARWorld training content can be described using the IEEE LTSC LOM specification, we will conform to the latest established standard for learning object metadata.

11.1.3 AICC

The Aviation Industry CBT (Computer-Based Training) Committee (AICC) is an international association focused on developing guidelines for the aviation industry in the areas of development, delivery, and evaluation of CBT and related training technologies. The stated objectives of the AICC[AICC] are:

- Assist airplane operators in development of guidelines, which promote the economic and effective implementation of computer-based training (CBT).
- Develop guidelines to enable interoperability.
- Provide an open forum for the discussion of CBT (and other) training technologies.

AICC was formed in 1988 to help standardize hardware for CBT delivery platforms and has since developed nine specifications (AICC Guidelines & Recommendations) or AGRs for learning, including:

AGR 002 - COURSEWARE DELIVERY STATIONS: which makes recommendations regarding CBT delivery systems including CPU, clock speed, bus, power supply, operating system, RAM, CD-ROM, graphic adapter, monitor, mouse, keyboard, digital audio system, videodisc player, and network.

AGR 003 - DIGITAL AUDIO: recommends guidelines that promote the interoperability of digital audio.

AGR 004 - OPERATING/WINDOWING SYSTEM: provides a formal recommendation for an operating and windowing system used for delivery of CBT.

AGR 005 - CBT PERIPHERAL DEVICES: recommends guidelines for interoperability of the following peripheral devices: video overlay card, videodisk player, and XY input device (such as a touch screen, mouse, or trackball), and part task trainers.

AGR 006 - COMPUTER-MANAGED INSTRUCTION: guidelines that promote the interoperability of CMI systems (on local file systems). Interoperability means the ability of a given CMI system to manage CBT lessons from different origins. It also includes the ability for a given CBT lesson to exchange data with different CMI systems.

AGR 007 - COURSEWARE INTERCHANGE: guidelines for the interchange of the elements that occur in CBT courseware. These elements include: Text, Graphics, Motion (frame-based), Audio, and Logic. Guidelines include: 1) major data components of CBT courseware, and 2) Standard data formats for those components.

AGR 008 - DIGITAL VIDEO: guidelines for the creation, distribution, and use of digital video in CBT courseware.

AGR 009 - ICON STANDARDS: USER INTERFACE: guidelines for the functions of the student/user interface and their associated graphic representation in CBT courseware.

AGR 010 - WEB-BASED COMPUTER-MANAGED INSTRUCTION: guidelines that promote the interoperability of web-based CMI systems. The purpose of this AGR is to promote the same kind of interoperability as described AGR006 for Web-based CBT courseware and CMI systems.

AICC, like have many e-learning organizations, has focused on reuse and interoperability of online learning. Because their history predates the World Wide Web, many of their standards have a decidedly different implicit architecture than today's architectures initially designed for the web and modern browser capabilities. AGR 010 – Web-based Computer Managed Instruction was designed to promote web-based interoperability. This communication API heavily influenced parts of the SCORM communication specification.

For DARWorld, the utility of AICC may be mostly in the capabilities supported by the AGRs and less in the actual architecture and implementation supporting those capabilities. AICCs coordination with other learning technology standards organizations such as LTSA, IMS and ADL will provide another avenue of utility for the DARWorld program.

11.1.4 IEEE LTSC

The IEEE 1484 Learning Technology Standards Committee (LTSC)[8] was formed in 1996 and is chartered by the IEEE Computer Society Standards Activity Board to develop technical standards, recommended practices and guides for a variety of aspects of learning technology including learning object metadata, course sequencing, student profiles, competency definitions, localization, and content packaging. IEEE LTSC is organized into a number of working groups and study groups including: Architecture and Reference Model Working Group (WG), Glossary WG, Computer Managed Instruction WG, Learning Objects Metadata WG, Semantics and Exchange Bindings WG, Data Interchange Protocols WG, Platform and Media Profiles WG, Competency Definitions WG, and Digital Rights Expression Language Study Group.

LTSC has developed a Learning Technology Systems Architecture (LTSA), shown in Figure 11, which specifies a high-level architecture for information technology-supported learning, education, and training systems. LTSA was designed to be pedagogically neutral, content-neutral, culturally neutral, and platform-neutral.

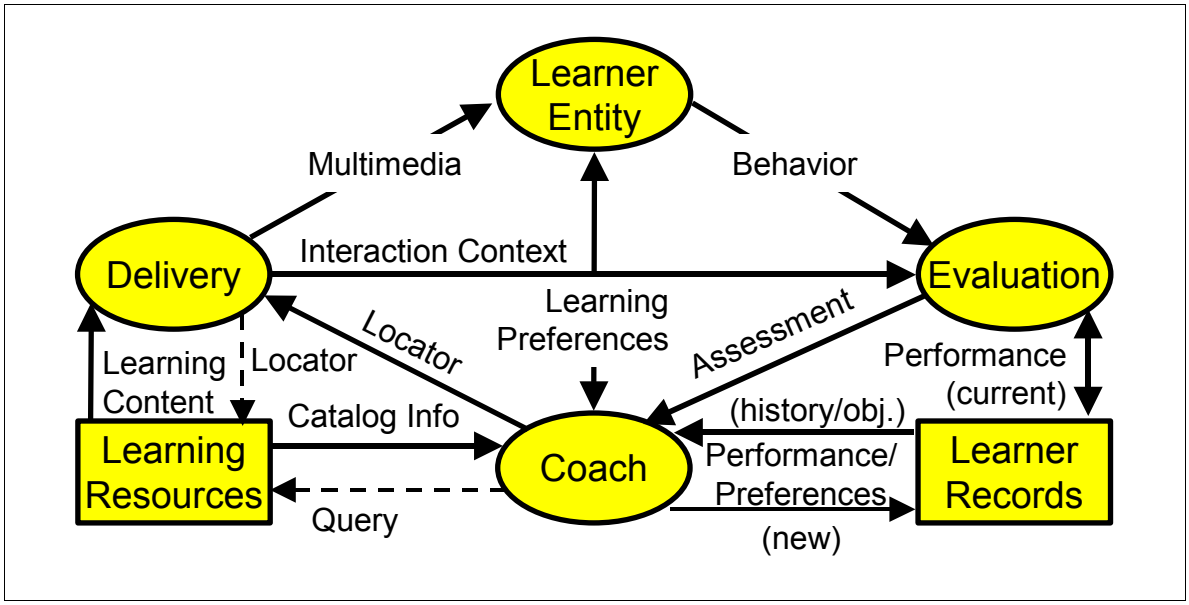

Figure 11: The LTSA System Components

The LTSA is organized around four processes: learner entity, evaluation, coach, and delivery process; two storage repositories: learner records and learning resources; and thirteen information flows among these components. The system components are designed to illustrate critical interoperability interfaces required by learning technology systems, but is not designed to show all interfaces for any particular learning technology system. The LTSA does not attempt to identify interoperability interfaces for related systems, such as content development or administrative systems. The LTSA merely identifies interfaces and does not specify any of the actual APIs, protocols, coding, etc. These specifications and standards are specifically outside the scope of the LTSA.

[8] http://ltsc.ieee.org

11.1.5 IMS Global Learning Consortium

The IMS (Instructional Management System) project is a coalition of corporate, academic, and government partners with the vision of creating a comprehensive open architecture and infrastructure for learning technologies.[IMS]

11.1.6 PROMETEUS

PROMETEUS (PROmoting Multimedia Access to Education and Training in EUropean Society) was started in 1999 and is sponsored by the European Commission. It is designed to help promote multimedia access to education and training throughout European society. Since 1999 it has evolved to encompass a wide range of technology-assisted learning interests. PROMETEUS is currently designed around a steering committee and nine special interest groups which are structured around the key areas for co-operation within the common Memorandum of Understanding (MoU), such as: Interchange of multimedia educational material, Knowledge and skills assessment, accreditation, Quality and Best Practice, Interoperability of services.

PROMETEUS has brought together hundreds of public and private sector organizations in Europe and thus deserves to be closely watched and coordinated with as they continue to progress and develop standards for learning technologies.

11.2 Acronyms

AAR	After Action Review
ACL	Access Control List
ADL	Advanced Distributed Learning Initiative
AFRL	Air Force Research Laboratory
AGR	AICC Guidelines & Recommendations (see AICC)
AICC	Aviation Industry CBT (Computer-Based Training) Committee
API	Application-Program Interface
ARIADNE	A European Union standards organization.
BEEP	Block Extensible Exchange Protocol
BLUEFOR	Blue Forces (friendly forces)
C2	Command & Control
CA	Certificate Authority
CBT	Computer-Based Training (see AICC)
CGF	Computer Generated Forces
CGI	Common Gateway Interface (web scripting)
CMI	Computer Managed Instruction
CONOPS	Concept of Operations
COTS	Commercial off-the-shelf
CRL	Certificate Revocation List (a type of server)
CTC	Combat Training Center (U.S. military)
DARPA	Defense Advanced Research Projects Agency
DIS	Distributed Interactive Simulation
DoD	Department of Defense
EJBCA	Enterprise Java Beans Certificate Authority
GOTS	Government off-the-shelf
HLA	High Level Architecture
HTTPS	Hypertext Transport Protocol with secure socket layer (SSL) encryption
ICT	Institute for Creative Technology
IDA	Institute for Defense Analyses
IDE	Integrated Development Environment
IEEE	Institute of Electrical & Electronics Engineers
IMS	Instructional Management System
ISI	Information Sciences Institute
ISO	International Organization for Standardization
ISR	Intelligence, Surveillance, and Reconnaissance – is this in the doc???
ITS	Intelligent Tutoring System
J2EE	Java 2 Enterprise Edition
Jabber, Jabberd	Jabber is an open, XML based, software platform. Jabberd provides a server implementation of the Jabber protocols.
Jabberoo	An object-oriented library for the Jabber protocol.
JDBC	Java Database Connectivity
JFCOM	Joint Forces Command

JNI	Java Native Interface
JSAF	Joint Semi Automated Force
JSO	Java Shared Object
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
LCMS	Learning Content Management System
LMS	Learning Management System
LMTS	Last Meter Training System
LOM	Learning Objects Metadata
LTSA	Learning Technology Systems Architecture
LTSC	Learning Technology Standards Committee
MAV	Micro Air Vehicle
MMOG	Massive Multiplayer Online Game
MMP	Massive Multi Player
ModSAF	Modular Semi Automated Force
MOUT	Military Operations on Urbanized Terrain
NPC	Non Player Character
OneSAF	(Army's single) Semi Automated Force
OOTS	Open source Off The Shelf
OPFOR	Opposing Forces
OPORD	Operational Orders
OSD	Office of the Secretary of Defense
PEO STRI	Program Executive Office for Simulation, Training & Instrumentation
PKI	Public Key Infrastructure
PROMETEUS	PROmoting Multimedia Access to Education and Training in EUropean Society
RA	Registration Authority
ROE	Rules Of Engagement
SAF	Semi Automated Force
SCORM	Sharable Content Object Reference Model
SIMNET	Simulation Network
SME	Subject Matter Expert
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
STRICOM	Simulation, Training, and Instrumentation Command
TLS	Transport Level Security
TSD	Training Session Descriptor – same as Training Package
UAV	Unmanned Aerial Vehicle
VoIP	Voice Over IP (Internet Protocol)
XML	eXtensible Markup Language
XMPP	Extensible Messaging and Presence Protocol
XMSF	Extensible Modeling and Simulation Framework
YAJA	Yet Another Jabber API

12 References

12.1 Related Documents

- DARWARS Developers' Guide
- DARWARS Integration and Transition Plan
- MÄK DIS/HLA Game-Link Developer's Guide

12.2 Related Links

[Apache] http://www.apache.org

[Ap.SSL] Apache-SSL http://www.apache-ssl.org

[modssl] http://www.modssl.org

[JNI] *Java Native Interface* http://java.sun.com/products/jdk/1.2/docs/guide/jni/

[SOAP] *Simple Object Access Protocol* http://www.w3.org/TR/SOAP/

[HTTPS] Hypertext Transport Protocol with secure socket layer (SSL) encryption

[BEEP] P. Resnick et al. *Block Extensible Exchange Protocol* http://www.ietf.org/html.charters/beep-charter.html RFC3117 etc. http://www.beepcore.org/beepcore/home.jsp

[PKI] Public Key Infrastructure

[EJBCA] http://ejbca.sourceforge.net

[Ariadne] http://www.ariadne-eu.org

[AICC] http://www.aicc.org

[IMS] http://www.ims.org